

TUGAS AKHIR - KI141502

**STUDI KINERJA METODE EKSTRAKSI FITUR
LOCAL LINE BINARY PATTERN DAN *SCALE
INVARIANT FEATURE TRANSFORM* PADA
APLIKASI *PALM* DAN *FINGER VEIN RECOGNITION***

LUSIANA NURUL AINI
5113100147

Dosen Pembimbing
Dr. Chastine Fatichah, S.Kom., M.Kom.
Bilqis Amaliah, S.Kom., M.Kom.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - KI141502

**STUDI KINERJA METODE EKSTRAKSI FITUR
LOCAL LINE BINARY PATTERN DAN *SCALE
INVARIANT FEATURE TRANSFORM* PADA
APLIKASI *PALM* DAN *FINGER VEIN RECOGNITION***

LUSIANA NURUL AINI
5113100147

Dosen Pembimbing I
Dr. Chastine Fatichah, S.Kom., M.Kom.

Dosen Pembimbing II
Bilqis Amaliah, S.Kom., M.Kom.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

PERFORMANCE STUDY OF FEATURE EXTRACTION METHOD OF LOCAL LINE BINARY PATTERN AND SCALE INVARIANT FEATURE TRANSFORM ON PALM AND FINGER VEIN RECOGNITION APPLICATION

LUSIANA NURUL AINI
5113100147

Dosen Pembimbing I
Dr. Chastine Fatichah, S.Kom., M.Kom.

Dosen Pembimbing II
Bilqis Amaliah, S.Kom., M.Kom.

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
Sepuluh Nopember Institute of Technology
Surabaya, 2017

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

STUDI KINERJA METODE EKSTRAKSI FITUR *LOCAL LINE BINARY PATTERN* DAN *SCALE INVARIANT FEATURE TRANSFORM* PADA APLIKASI *PALM* DAN *FINGER VEIN RECOGNITION*

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Komputasi Cerdas dan Visi
Program Studi S-1 Departemen Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

LUSIANA NURUL AINI

NRP: 5113 100 147

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Dr. Chastine Fatichah, S.Kom., M.Kom.
NIP. 197512202001122002



(Pembimbing 1)

Bilqis Amaliah, S.Kom., M.Kom.
NIP. 197509142001122002

(Pembimbing 2)

**SURABAYA
JULI, 2017**

[Halaman ini sengaja dikosongkan]

STUDI KINERJA METODE EKSTRAKSI FITUR *LOCAL LINE BINARY PATTERN* DAN *SCALE INVARIANT FEATURE TRANSFORM* PADA APLIKASI *PALM* DAN *FINGER VEIN RECOGNITION*

Nama Mahasiswa : Lusiana Nurul Aini
NRP : 5113 100 147
Jurusan : Teknik Informatika, FTIf ITS
Dosen Pembimbing 1 : Dr. Chastine Fatichah, S.Kom., M.Kom.
Dosen Pembimbing 2 : Bilqis Amaliah, S.Kom., M.Kom.

Abstrak

Dalam perkembangan teknologi, sosial, maupun ekonomi saat ini validasi untuk keamanan informasi telah meningkat. Namun metode otentikasi tradisional seperti password dan kunci tidak dapat memenuhi harapan karena sangat rentan untuk dilupakan, hilang ataupun dicuri. Dengan adanya hal tersebut maka berkembanglah identifikasi dan otentikasi berbasis karakteristik biometrik yang tidak mungkin dilupakan maupun dipalsu yaitu metode otentikasi biometrik menggunakan teknik pengenalan pola pembuluh darah di jari (Finger Vein) maupun telapak tangan (Palm Vein). Untuk melakukan pengenalan pembuluh darah setiap individu, akan dilakukan proses ekstraksi fitur atau mendapatkan ciri dari pembuluh darah tersebut. Proses ekstraksi fitur ini sangat berpengaruh dalam tahap pengenalan pola setiap individu.

Tugas akhir ini mengusulkan perbandingan metode ekstraksi fitur menggunakan metode Local Line Binary Pattern (LLBP) dan Scale Invariant Feature Transform (SIFT). Tujuan dari tugas akhir ini yaitu untuk mengetahui performa antara dua metode tersebut dalam melakukan pengenalan pola pembuluh darah pada palm maupun finger vein. Data masukan berupa data testing dari palm dan finger vein, dan data training untuk mendapatkan akurasi dari setiap uji coba metode. Proses pengenalan pola pembuluh darah terdapat 3 proses utama yaitu

preprocessing yang bertujuan untuk memperbaiki citra agar terlihat pola pembuluh darah, kemudian ekstraksi fitur untuk mendapatkan fitur dari pola pembuluh darah, dan tahap terakhir yaitu melakukan similarity antara data testing dengan data training. Sehingga akan mendapatkan hasil akurasi setiap metode yang diimplementasikan pada setiap dataset.

Data yang digunakan untuk uji coba yaitu 35 individu data hasil scan palm dan finger vein. Untuk rata-rata hasil akurasi uji coba pengenalan palm vein menggunakan metode ekstraksi fitur LLBP yaitu 85.72%. Sedangkan untuk pengenalan finger vein menggunakan metode LLBP yaitu 56.57%. Untuk rata-rata akurasi pada pengenalan palm vein menggunakan metode SIFT adalah 58.93%. Uji coba yang terakhir yaitu untuk pengenalan finger vein menggunakan metode SIFT mempunyai rata-rata akurasi sebesar 80.72%. Dari hasil uji coba dapat disimpulkan bahwa metode LLBP lebih cocok digunakan untuk pengenalan palm vein, sedangkan metode SIFT lebih cocok digunakan untuk pengenalan finger vein.

Kata kunci: Local Line Binary Pattern (LLBP), Scale Invariant Feature Transform (SIFT), palm vein, finger vein, similarity, matching, operator LLBP, threshold, pembuluh darah.

PERFORMANCE STUDY OF FEATURE EXTRACTION METHOD OF LOCAL LINE BINARY PATTERN AND SCALE INVARIANT FEATURE TRANSFORM ON PALM AND FINGER VEIN RECOGNITION APPLICATION

Student Name : Lusiana Nurul Aini
Registration Number : 5113 100 147
Department : Informatics Engineering, FTIf ITS
First Supervisor : Dr. Chastine Fatichah, S.Kom., M.Kom.
Second Supervisor : Bilqis Amaliah, S.Kom., M.Kom.

Abstract

In current technological, social, and economic developments the validation for information security has increased. However, traditional authentication methods such as passwords and keys can not meet expectations because they are vulnerable to being forgotten, lost or stolen. Given this, it will be able to identify and authenticate based on biometric characteristics that can not be forgotten or falsified, namely biometric authentication method using vein pattern recognition technique on fingers (Finger Vein) and palms (Palm Vein). To perform the introduction of individual blood vessels, a feature extraction process or characteristic of the blood vessels will be performed. This feature extraction process is very influential in the pattern recognition stage of each individual.

In this final project proposes to compare feature extraction method using Local Line Binary Pattern (LLBP) method and Scale Invariant Feature Transform (SIFT). The purpose of this final task is to determine the performance between the two methods in performing pattern recognition of blood vessels in the palm and finger vein. The input data are data testing from palm and finger vein, and training data to get accurate from each test method. Vocabulary pattern recognition process consists of 3 main processes: preprocessing aimed at improving the image for vascular patterning, then feature extraction to get feature of blood

vessel pattern, and the last stage is to do similarity between data testing and training data. So that will get the accuracy of each method implemented on each dataset.

The data used for the experiments are 35 individual data of palm scan and finger vein. For the average of accuracy test results of the introduction of palm vein using LLBP feature extraction method is 85.72%. As for the introduction of finger vein using LLBP method is 56.57%. For the average accuracy on the introduction of palm vein using the SIFT method is 58.93%. The last test that is for the introduction of finger vein using SIFT method has an average accuracy of 80.72%. From the experimental result it can be concluded that LLBP method is more suitable for the introduction of palm vein, while SIFT method is more suitable for finger vein recognition.

Keywords: *Local Line Binary Pattern (LLBP), Scale Invariant Feature Transform (SIFT), palm vein, finger vein, similarity, matching, LLBP operator, threshold, blood vessel.*

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **“Studi Kinerja Metode Ekstraksi Fitur Local Line Binary Pattern dan Scale Invariant Feature Transform pada Aplikasi *Palm* dan *Finger Vein Recognition*”**.

Buku tugas akhir ini disusun dengan harapan dapat memberikan manfaat dalam penelitian studi kinerja metode ekstraksi fitur LLBP dan SIFT lebih lanjut. Selain itu, penulis berharap dapat memberikan kontribusi positif bagi kampus Teknik Informatika ITS.

Dalam perancangan, pengerjaan, dan penyusunan tugas akhir ini, penulis banyak mendapatkan bantuan dari berbagai pihak. Penulis ingin mengucapkan terima kasih kepada:

1. Ibu Dr. Chastine Fatichah, S.Kom., M.Kom. dan Ibu Bilqis Amaliah, S.Kom., M.Kom. selaku dosen pembimbing penulis yang telah memberi ide, nasihat dan arahan sehingga penulis dapat menyelesaikan tugas akhir dengan tepat waktu.
2. Orang tua penulis Bapak Ma'sum Sa'dullah dan Ibu Siti Masruroh yang telah memberikan dukungan moral, spiritual dan material serta senantiasa memberikan doa demi kelancaran dan kemudahan penulis dalam mengerjakan tugas akhir.
3. Satu-satunya saudara kandung saya yaitu Alf Rahmat, seluruh keluarga besar, dan calon pendamping hidup penulis yang telah memberikan dukungan yang besar baik secara langsung maupun secara implisit.
4. Teman-teman di Lab KCV: para admin yang telah banyak membantu memfasilitasi dalam pengerjaan serta teman-

teman yang berhabitat hidup di KCV, yang sudah menemani penulis.

5. Sahabat-sahabat saya (Dhea, Hanif, Sani, Rere, Eriko, Nuning, Ine, Yohana) yang telah memberi semangat dan motivasi untuk mengerjakan, serta teman satu bimbingan Bu Chastine (Nanang) yang selalu memberi informasi mengenai bimbingan.
6. Teman admin Laboratorium LP2 (John Stephanus P) yang telah membantu penulis dalam mengerjakan tugas akhir ini dan memberi motivasi untuk tidak patah semangat.
7. Teman-teman seperti Ekky, Ayu, Nida, Fikri, Dhanar, dan mahasiswa angkatan 2013 lain yang sama-sama mengarungi empat tahun masa perkuliahan bersama penulis, yang saat berinteraksi dan diam-diam mengamati tingkah laku mereka, telah memberi penulis banyak pelajaran hidup secara tersirat maupun tersurat dengan gratis.
8. Pihak-pihak lain yang tidak bisa penulis sebutkan satu-persatu.

Penulis menyadari masih ada kekurangan dalam penyusunan tugas akhir ini. Penulis mohon maaf atas kesalahan, kelalaian maupun kekurangan dalam penyusunan tugas akhir ini. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan ke depan.

Surabaya, Juli 2017

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	Error! Bookmark not defined.
Abstrak	vii
Abstract	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR KODE SUMBER	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Tugas Akhir	3
1.5 Manfaat Tugas Akhir	3
1.6 Metodologi	4
1.7 Sistematika Laporan.....	5
BAB II DASAR TEORI.....	7
2.1 Pengenalan Palm Vein	7
2.2 Pengenalan Finger Vein	7
2.3 Local Line Binary Pattern (LLBP).....	8
2.4 Scale Invariant Feature Transform (SIFT)	10
2.5 Region of Interest (ROI)	12
2.6 Adapative Histogram Equalization	13
2.7 Adaptive Noise Removal	15
2.8 Anisotropic Diffusion Filter	16
2.9 Frangi Filter.....	17
2.10 Median Filter.....	18
2.11 Operasi Morfologi.....	19
2.12 Similarity.....	22
BAB III ANALISIS DAN PERANCANGAN	23
3.1 Tahap Analisis.....	23
3.1.1 Deskripsi Umum.....	23

3.1.2 Spesifikasi Kebutuhan Sistem	23
3.1.3 Analisis Permasalahan	24
3.2 Tahap Perancangan	31
3.2.1 Perancangan Sistem	31
3.2.2 Perancangan Data	32
3.2.3 Perancangan Proses	33
BAB IV IMPLEMENTASI.....	43
4.1 Lingkungan Implementasi.....	43
4.1.1 Perangkat Keras	43
4.1.2 Perangkat Lunak	43
4.2 Implementasi Tahap Preprocessing.....	44
4.2.1 Pada Palm Vein.....	44
4.2.2 Pada Finger Vein	51
4.3 Implementasi Ekstraksi Fitur	52
4.3.1 Implementasi Ekstraksi Fitur LLBP	52
4.3.2 Implementasi Ekstraksi Fitur SIFT	55
4.3.3 Implementasi Similarity pada Palm Vein	55
4.3.4 Implementasi Matching pada Finger Vein.....	56
BAB V UJI COBA DAN EVALUASI.....	57
5.1 Lingkungan Uji Coba	57
5.2 Data Uji Coba.....	57
5.3 Skenario Uji Coba	59
5.4 Uji Coba Mendapatkan Akurasi pada Palm Vein Recognition menggunakan Metode Ekstraksi Fitur LLBP.....	60
5.5 Uji Coba Mendapatkan Akurasi pada Finger Vein menggunakan Metode Ekstraksi Fitur LLBP	61
5.6 Uji Coba Mendapatkan Akurasi pada Palm Vein menggunakan Metode Ekstraksi Fitur SIFT.....	62
5.7 Uji Coba Mendapatkan Akurasi pada Finger Vein menggunakan Metode Ekstraksi Fitur SIFT.....	64
5.8 Evaluasi	65
5.8.1 Evaluasi pada Tahap Preprocessing.....	65
5.8.2 Evaluasi pada Tahap Ekstraksi Fitur	65

5.8.3 Evaluasi pada Tahap Matching untuk data Finger Vein....	68
BAB VI KESIMPULAN DAN SARAN.....	69
6.1 Kesimpulan	69
6.2 Saran	70
BIODATA PENULIS.....	81

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Contoh Implementasi Metode LLBP [1]	9
Gambar 2.2 Pembentukan deskriptor keypoint	12
Gambar 2.3 Adaptive Histogram Equalization	14
Gambar 2.4 Citra sebelum dilakukan adapthisteq [7]	14
Gambar 2.5 Citra setelah dilakukan adapthisteq [7]	15
Gambar 2.6 Citra Sebelum dilakukan filter Adaptive Noise Removal [9]	16
Gambar 2.7 Citra Setelah dilakukan filter Adaptive Noise Removal [9]	16
Gambar 2.8 Citra Sebelum dilakukan Median Filter [10]	19
Gambar 2.9 Citra Setelah dilakukan Median Filter [10]	19
Gambar 2.10 Citra Sebelum dilakukan Morfologi Closing [11]	21
Gambar 2.11 Citra Setelah dilakukan Morfologi Closing [11]	21
Gambar 2.12 Citra Sebelum dilakukan Morfologi Opening [12]	21
Gambar 2.13 Citra Setelah dilakukan Morfologi Opening [12]	22
Gambar 3.1 Perbedaan rotasi tangan	25
Gambar 3.2 Perbedaan Intensitas Cahaya	26
Gambar 3.3 Hasil Ekstraksi Citra Palm Vein dengan panjang operator LLBP yaitu 11	27
Gambar 3.4 Hasil Ekstraksi Citra Palm Vein dengan panjang operator LLBP yaitu 17	27
Gambar 3.5 Hasil Keypoint Individu 1 pada Pengambilan Ketiga	28
Gambar 3.6 Hasil Keypoint Individu 1 pada Pengambilan Keempat	28
Gambar 3.7 Cropping diluar border jari	29
Gambar 3.8 Cropping didalam border jari	29
Gambar 3.9 Citra scan palm vein	31
Gambar 3.10 Citra scan finger vein	31
Gambar 3.11 Diagram Alir dari Sistem Perbandingan Metode Ekstraksi Fitur pada Palm Vein dan Finger Vein	33
Gambar 3.12 Citra hasil proses Region of Interest	34
Gambar 3.13 Diagram Alir Proses untuk Mendapatkan Region of Interest dari Palm Vein	35

Gambar 3.14 Diagram Alir Tahap Preprocessing pada Palm Vein	36
Gambar 3.15 Citra Preprocessing (a)Hasil Resize (b)Median Filter (c)Adaptive Histeq (d)Noise Removal (e)ADF (f)Convert (g)Closing (h)Subtract (i)Adjust	37
Gambar 3.16 Citra Preprocessing (a)Citra Asli (b)Adaptive Histeq (c)Frangi Filter (d)Median Filter (e)Closing (f)Hasil Cropping ..	38
Gambar 3.17 Diagram Alir Tahap Preprocessing pada Finger Vein	39
Gambar 3.18 Diagram Alir Tahap Ekstraksi Fitur LLBP.....	39
Gambar 3.19 Citra Hasil Ekstraksi Fitur LLBP.....	40
Gambar 3.20 Diagram Alir Tahap Ekstraksi Fitur SIFT	40
Gambar 3.21 Hasil Ekstraksi Fitur menggunakan metode SIFT.	41
Gambar 5.1 Citra Hasil Scan Palm Vein	58
Gambar 5.2 Citra Hasil Proses ROI Detection	58
Gambar 5.3 Citra Hasil Scan Finger Vein	59
Gambar 5.4 Citra Hasil Proses Cropping	59
Gambar 5.5 Citra Hasil Ekstraksi fitur LLBP (a) Dengan N=11 (b) N=13 (c) N=15 (d) N=17 (e) N=19	61
Gambar 5.6 Representasi Fitur LLBP dengan Panjang Operator LLBP 11	66
Gambar 5.7 Matriks Hasil Ekstraksi Fitur LLBP dengan Panjang Operator 11	66
Gambar 5.8 Fitur SIFT (a)Keypoint (b)Deskriptor	67

DAFTAR TABEL

Tabel 5.1 Hasil Uji Coba Metode LLBP pada Palm Vein.....	60
Tabel 5.2 Hasil Uji Coba Metode LLBP pada Finger Vein	62
Tabel 5.3 Hasil Uji Coba Metode SIFT pada Palm Vein	63
Tabel 5.4 Hasil Uji Coba Metode SIFT pada Finger Vein.....	64

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi ROI <i>Detection</i>	49
Kode Sumber 4.2 Implementasi Pencarian <i>fpeak</i> pada Proses ROI <i>Detection</i>	50
Kode Sumber 4.3 Implementasi <i>Similarity</i> pada metode LLBP .	56
Kode Sumber 4.4 Implementasi <i>Matching</i> pada Metode SIFT ...	56

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini dibahas hal-hal yang mendasari tugas akhir. Bahasan meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika laporan tugas akhir.

1.1 Latar Belakang

Dalam perkembangan teknologi, sosial, maupun ekonomi saat ini validasi untuk keamanan informasi telah meningkat. Namun metode otentikasi tradisional seperti password dan kunci tidak dapat memenuhi harapan karena sangat rentan untuk dilupakan, hilang ataupun dicuri. Dengan adanya hal tersebut maka berkembanglah identifikasi dan otentikasi berbasis karakteristik biometrik yang aman, nyaman dan tidak mungkin dilupakan maupun dipalsu yaitu metode otentikasi biometrik yang menggunakan teknik pengenalan pola berdasarkan pola pembuluh darah vena di jari (*Finger Vein*) maupun telapak tangan (*Palm Vein*) manusia yang ada di bawah permukaan kulit [1].

Selain *finger vein* dan *palm vein* ada beberapa jenis kewanan informasi maupun data lainnya seperti *face recognition*, *fingerprint*, *speak recognition*, *iris scan*, dll. Namun metode-metode tersebut masih belum sempurna dalam semua sekenario aplikasi [2]. Sebagai contoh sistem pengenalan wajah (*face recognition*) yang user friendly tetapi sistem tersebut lebih rentan terhadap kesalahan individu ketika mengidentifikasi dengan sudut pandang yang berbeda atau ekspresi yang berbeda. Kemudian bisa juga dipengaruhi oleh topi, rambut, mata, jenggot, dan sebagainya [2]. Sedangkan untuk *fingerprint* rentan terhadap reproduksi sehingga mempengaruhi pengenalan sidik jari. Untuk mengatasi masalah tersebut maka dari itu berkembanglah teknik otentikasi baru dalam mengidentifikasi setiap individu yaitu melalui pengenalan pembuluh darah vena pada jari maupun telapak tangan setiap individu [3]. Metode pengenalan baru ini membedakan

jaringan pembuluh darah vena antar individu dan sulit untuk dipalsukan.

Finger vein maupun *palm vein* melibatkan dua tahap yaitu Pendaftaran dan Verifikasi. Dalam tahap pendaftaran, pembuluh darah vena (vena) dari setiap individu ditangkap yang biasanya menggunakan terminal attester mengandung inframerah dengan cahaya LED dan CCD, yang kemudian akan muncul sebagai pola gelap garis [2]. Setelah itu, pengolahan citra metode yang digunakan untuk meningkatkan kualitas gambar vena. Kemudian pada tahap verifikasi, gambar vena dilakukan *preprocessing* citra dan ekstraksi fitur. Akhirnya, hasil verifikasi diperoleh dengan membandingkan kesamaan antara fitur pengujian dengan yang data fitur yang telah disimpan. Karena pencocokan terakhir tergantung pada ekstraksi fitur, sehingga ekstraksi fitur merupakan tahap yang penting dalam pengenalan vena.

Pada tahap ekstraksi fitur ada beberapa metode yang bisa digunakan, pada usulan tugas akhir ini penulis akan melakukan percobaan ekstraksi fitur pada *palm vein* dan *finger vein* menggunakan metode SIFT (*Scale Invariant Feature transform*) dan LLBP (*Local Line Binary Pattern*). Penulis akan membandingkan metode SIFT dan LLBP yang akan diterapkan pada *palm* dan *finger vein*, lebih bagus manakah diantara SIFT dan LLBP untuk *palm* maupun *finger vein*.

Metode ekstraksi fitur SIFT telah digunakan antara lain untuk kasus pembuluh darah telapak tangan, pembuluh darah jari, dan pengenalan wajah. Dari penelitian tersebut metode SIFT teruji memberikan nilai kesalahan pengenalan yang rendah. Meskipun kemampuan SIFT dapat diandalkan, tetapi ada beberapa faktor yang mempengaruhi hasil pengenalan, diantaranya yaitu perubahan skala, rotasi, translasi, dan perbedaan intensitas cahaya. Hal ini dikarenakan fitur SIFT sangat bergantung pada hasil penentuan *keypoint*, sedangkan jumlah *keypoint* dapat dipengaruhi oleh intensitas cahaya [2]. Sedangkan untuk metode LLBP dapat mengekstrak fitur dari gambar dengan pembuluh darah yang

kurang jelas sehingga dapat menangkap pola pada dalam pembuluh darah tersebut [1].

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana melakukan ekstraksi fitur *palm* dan *finger vein* menggunakan metode LLBP?
2. Bagaimana melakukan ekstraksi fitur *palm* dan *finger vein* menggunakan metode SIFT?
3. Bagaimana performa metode SIFT dan LLBP yang diterapkan pada *palm* dan *finger vein recognition*?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain:

1. Metode untuk ekstraksi fitur yang digunakan SIFT dan LLBP.
2. Percobaan akan dilakukan menggunakan MATLAB.
3. Data citra yang digunakan yaitu data citra yang terdapat pada database CASIA untuk *palm vein* dan *Hongkong Polytechnic University* (Kumar&Zhou, 2012) untuk data citra *finger vein*.

1.4 Tujuan Tugas Akhir

Tujuan dari pembuatan tugas akhir ini yaitu untuk mengetahui performa pada metode ekstraksi fitur yang menghasilkan identifikasi lebih bagus untuk pengenalan *palm* dan *finger vein*.

1.5 Manfaat Tugas Akhir

Manfaat dari pembuatan tugas akhir ini yaitu mengetahui metode pada ekstraksi fitur yang memiliki tingkat akurasi tinggi.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. Studi Literatur

Studi literatur yang dilakukan dalam pengerjaan Tugas Akhir ini adalah mengenai ROI *detection*, ekstraksi fitur pada *palm vein* dan *finger vein* menggunakan metode SIFT dan LLBP, *feature matching* pada *palm vein*.

2. Analisis dan Desain Perangkat Lunak

Rencana perangkat lunak yang digunakan dibangun menggunakan Matlab untuk melakukan identifikasi *palm vein* dan *finger vein* seseorang yang menghasilkan *output* akurasi. *Input* yang diterima berupa citra data testing yang dipilih akan diproses oleh aplikasi dan akan melakukan proses *recognition* dan akan menghasilkan nilai akurasi setiap percobaan.

3. Implementasi Perangkat Lunak

Implementasi untuk melakukan percobaan ini adalah menggunakan Matlab, mulai dari proses *palm vein recognition* dan *finger vein recognition*.

4. Uji Coba dan Evaluasi

Metode Analisis Fitur:

- Ekstraksi fitur pada *palm vein recognition* dan *finger vein recognition* menggunakan metode SIFT maupun LLBP.
- *Feature matching* pada setiap metode.
- Hitung akurasi dengan uji coba pada *dataset*.

Implementasi:

- *Tools* yang digunakan adalah Matlab.

Pengujian:

- Memilih data testing yang akan digunakan uji coba, data testing berupa citra yang telah dilakukan *preprocessing*.

- Ekstraksi fitur pada data *test* dan data *training* menggunakan metode SIFT dan LLBP, kemudian proses *recognition* data *test* dengan data *training*.
- Sistem akan mengeluarkan hasil akurasi jika menggunakan metode SIFT dan LLBP.

1.7 Sistematika Laporan

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut:

Bab I Pendahuluan

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan tugas akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Dasar Teori

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan tugas akhir ini.

Bab III Analisis dan Perancangan

Bab ini berisi tentang analisis dan perancangan desain aplikasi perbandingan metode ekstraksi fitur pada *palm* dan *finger vein recognition*.

Bab IV Implementasi

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa kode yang digunakan untuk proses implementasi.

Bab V Uji Coba dan Evaluasi

Bab ini membahas tahap-tahap uji coba. Kemudian hasil uji coba dievaluasi untuk kinerja dari aplikasi yang dibangun.

Bab VI Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan aplikasi ke depannya.

BAB II

DASAR TEORI

Pada bab ini diuraikan mengenai dasar-dasar teori yang digunakan dalam pengerjaan tugas akhir dengan tujuan untuk memberikan gambaran secara umum terhadap penelitian yang dikerjakan. Bab ini berisi penjelasan mengenai pengenalan *palm vein*, pengenalan *finger vein*, ekstraksi fitur Local Line Binary Pattern, ekstraksi fitur Scale Invariant Feature Transform, praproses pada *palm vein*, praproses pada *finger vein*, dan algoritma *similarity* antar gambar.

2.1 Pengenalan Palm Vein

Metode otentikasi biometrik yang menggunakan teknik pengenalan pola berdasarkan pola pembuluh darah vena di telapak tangan manusia di bawah permukaan kulit. Pembuluh darah vena yang terletak pada telapak tangan dianggap sebagai bagian yang sesuai dengan pengenalan biometrik. Pembuluh darah vena yang paling mudah diakuisisi terletak pada bagian telapak tangan karena pada bagian telapak tangan tidak terdapat bulu yang dapat mengganggu proses akuisisi dan bagian tersebut tidak dipengaruhi oleh warna kulit.

Proses akuisisi *palm vein* menggunakan sensor yang terdiri dari *illumination device* yang menghasilkan sinar inframerah yang akan menyinari telapak tangan dan *capturing device* yang dapat berupa kamera digital tanpa filter inframerah atau kamera *digital noise reduction* (DNR) [4].

2.2 Pengenalan Finger Vein

Metode otentikasi biometrik yang menggunakan teknik pengenalan pola berdasarkan pola pembuluh darah di jari manusia di bawah permukaan kulit. Pengenalan *Finger Vein* ini digunakan untuk mengidentifikasi dan memverifikasi identitas individu. Untuk pengambilan citra pembuluh darah jari menggunakan sinar inframerah jarak dekat dengan panjang gelombang antara 700-

1000 nanometer yang dapat menyerap secara intensif kandungan hemoglobin dalam darah, akan tetapi jaringan lain akan mudah terambil sehingga pola dari pembuluh darah akan ditangkap sebagai bayangan [2].

2.3 Local Line Binary Pattern (LLBP)

LLBP merupakan metode pengembangan dari LBP (*Local Binary Pattern*), metode yang menggunakan kombinasi nilai-nilai biner dari pixel ketetanggaannya. Operator dari metode LLBP terdiri dari 2 komponen yaitu horizontal LLBP (LLBPh), vertical LLBP (LLBPv), dan *magnitude* dari LLBP. Nilai LLBP bisa diperoleh dari perhitungan *line binary code* dari kedua komponen [1]. Threshold $s(x)$ untuk pixel di dekatnya ini adalah nilai pusat pixel dan hasil dianggap sebagai biner. Threshold pixel didekatnya bisa diperoleh dari persamaan 2.1.

$$S(x) = 1, \text{ if } x \geq 0 ; 0, \text{ if } x < 0 \quad (2.1)$$

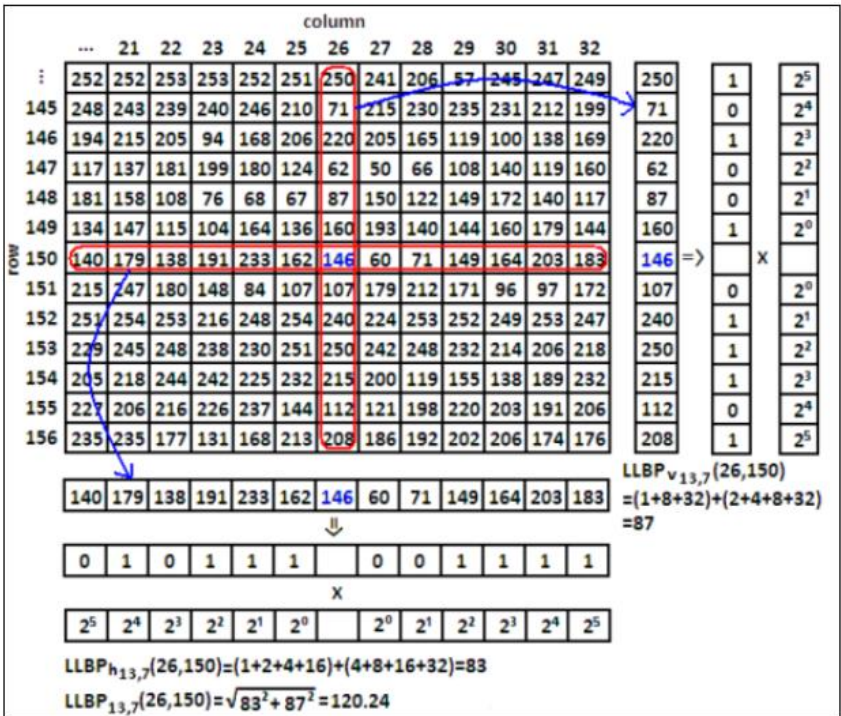
N adalah panjang barisan pixel ketetanggaan yang selanjutnya akan digunakan untuk parameter pada uji coba. $c = N/2$ adalah posisi pixel tengah. h_n merupakan piksel pada garis horizontal, v_n merupakan piksel pada garis vertikal, dan h_c merupakan piksel pada posisi tengah.

Sebuah komponen horizontal LLBP mengekstrak sebuah kode biner dari $N-1$ bit untuk setiap pixel menggunakan persamaan $s(x)$ dengan persamaan 2.2.

$$LLBP_{hN,c}(x, y) = \sum_{n=1}^{c-1} s(h_n - h_c) \cdot 2^{c-n-1} + \sum_{n=c+1}^N s(h_n - h_c) \cdot 2^{c-n-1} \quad (2.2)$$

Sedangkan nomor yang sama dari bit yang diekstrak oleh LLBP vertikal menggunakan persamaan $s(x)$ pada persamaan 4 dan persamaan 2.3.

$$LLBP_{vN,c}(x,y) = \sum_{n=1}^{c-1} s(v_n - v_c) \cdot 2^{c-n-1} + \sum_{n=c+1}^N s(v_n - v_c) \cdot 2^{c-n-1} \quad (2.3)$$



Gambar 2.1 Contoh Implementasi Metode LLBP [1]

Dari hasil perhitungan pada persamaan 2.2 dan 2.3, akan didapatkan nilai *magnitude* LLBP pada persamaan 2.4.

$$LLBP_m = \sqrt{LLBP_h^2 + LLBP_v^2} \quad (2.4)$$

Ilustrasi operator LLBP dengan $N=13$ ditunjukkan pada **Gambar 2.1** dengan penjelasan rumus pada persamaan 2.1, 2.2, dan 2.2. Pembobotan kode *binary* dimulai dari pixel kiri dan kanan dari pixel terdekat untuk horizontal, begitu juga dimulai dari pixel atas dan bawah terdekat untuk vertikal. Batas pixel paling ujung sesuai dengan nilai N yang ditentukan. Nilai *magnitude* LLBP yang dihasilkan dari persamaan 2.4 merupakan fitur LLBP dari pixel tersebut.

2.4 Scale Invariant Feature Transform (SIFT)

Fitur SIFT pertama kali diperkenalkan oleh Lowe pada tahun 2004, untuk memberikan hasil pencocokan citra yang terpercaya dan memperbaiki kinerja objek pada kondisi citra dengan efek perubahan skala, rotasi, dan translasi [5]. Langkah-langkah untuk memperoleh fitur SIFT sebagai berikut:

1. Mendeteksi *scal-space extrema* dan *keypoint* menggunakan *Difference of Gaussian* (DoG) sesuai dengan persamaan 2.5, dimana merupakan perhitungan dari selisih antara ruang skala yang terdekat. Ruang skala citra didefinisikan dengan $L(x, y, \sigma)$, yang mempunyai citra masukan $I(x, y)$ dan faktor skala σ . Maka ruang skala diperoleh dari hasil konvolusi dengan skala variabel Gaussian $G(x, y, \sigma)$ sesuai dengan persamaan 2.6 dan 2.7.

$$D(x, y, \sigma) = L(x, y, \sigma_1) - L(x, y, \sigma_2) \quad (2.5)$$

$$L(x, y, \sigma) = G(x, y, \sigma).I(x, y) \quad (2.6)$$

$$G(x, y, \sigma) = \frac{1}{2\pi^2} e^{-(x^2+y^2)/2\sigma^2} \quad (2.7)$$

2. Menentukan lokasi dan skala dari setiap kandidat *keypoint* yang telah diperoleh dari tahap sebelumnya. Kandidat *keypoint* yang memiliki kontras rendah, sensitif terhadap noise, dan terletak di sekitar tepi dapat dieliminasi. Untuk menghilangkan pengaruh bagian tepi, dilakukan penghitungan menggunakan *Hessian matrix* dengan berukuran 2×2 , seperti persamaan 2.9. Sedangkan standar pengukuran stabilitas *keypoints* untuk menentukan suatu kandidat *keypoint* dihilangkan atau tidak maka mengikuti persamaan 2.8 dengan nilai r menyatakan perbandingan nilai *eigenvalue* terbesar dan *eigenvalue* terkecil.

$$\frac{(D_{xx} + D_{yy})^2}{D_{xx}D_{yy} - D_{xy}^2} = \frac{(r + 1)^2}{r} \quad (2.8)$$

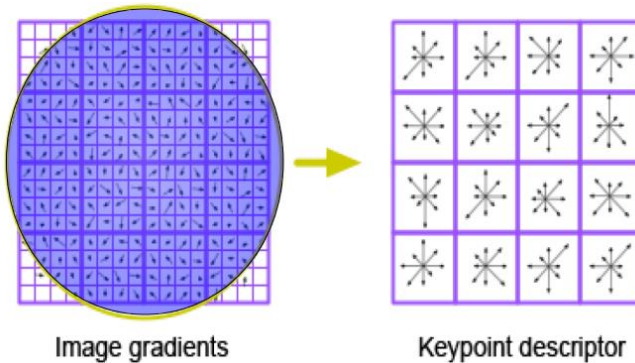
$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (2.9)$$

3. Menentukan arah orientasi dari setiap *keypoint* yang telah diperoleh dengan didasarkan pada arah gradien lokal citra. Besar gradien $m(x, y)$ dan orientasinya $\theta(x, y)$ diperoleh dari hasil perhitungan pixel-pixel pada setiap contoh citra $L(x, y)$ yang mengikuti persamaan 2.10 dan 2.11.

$$m(x, y) = \sqrt{\frac{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2}{2}} \quad (2.10)$$

$$\theta(x, y) = \tan^{-1} \left(\frac{L(x + 1, y) - L(x - 1, y)}{L(x, y + 1) - L(x, y - 1)} \right) \quad (2.11)$$

4. Menentukan deskriptor untuk *keypoint* dengan menggunakan histogram dari orientasi. Deskriptor untuk setiap *keypoint* yang terentuk dirangkum dalam sub wilayah berukuran 4×4 dimana setiap sub wilayah menyimpan informasi orientasi dalam delapan arah. **Gambar 2.2** menunjukkan pembentukan deskriptor *keypoint*. Setiap *keypoint* memiliki panjang vektor fitur 128 bit dengan 8 bit histogram yang digunakan.



Gambar 2.2 Pembentukan deskriptor *keypoint*

2.5 Region of Interest (ROI)

Proses pengambilan ROI dilakukan pada dataset *palm vein* menggunakan algoritma *inner border tracing*. Algoritma ini digunakan untuk membentuk citra border dari suatu objek. Pada tugas akhir ini, algoritma *inner border tracing* menggunakan 8-*connectivity* yang artinya akan mencari pixel tetangga dari 8 arah. Inisialisasi untuk mencari posisi pixel tetangga pada 8-*connectivity* yaitu 7. Untuk mendapatkan nilai arah dari pixel tetangga menggunakan rumusan $(\text{posisi pixel} + 6) \bmod 8$, jika nilai dari arah pencarian sebelumnya merupakan bilangan genap. Sedangkan jika nilai arah pencarian sebelumnya adalah bilangan ganjil, maka menggunakan rumusan $(\text{posisi pixel} + 6) \bmod 8$ [6].

Dalam mencari nilai arah baru hanya perlu mengikuti arah berlawanan dengan jarum jam. Untuk menghitung jarak pada

setiap pixel border menuju ke pixel pada bagian tengah bawah citra menggunakan diagram distribusi jarak. Diagram ini memiliki bentuk yang sama dengan telapak tangan, karena diagram ini memiliki lima puncak dan empat lembah dimana tiap puncak mewakili jari pada tiap citra input. Tetapi ada perbedaan dengan citra input, jari paling kiri pada diagram adalah jari kelingking, sedangkan pada citra input jari paling kiri adalah ibu jari. Perbedaan ini disebabkan karena proses pada algoritma *inner border tracing* memiliki arah berlawanan dengan jarum jam.

Dengan adanya diagram distribusi jarak ini akan didapatkan titik P_1 dan P_2 . Dimana P_1 adalah lembah yang berada diantara jari kelingking dan jari manis, sedangkan P_2 adalah titik lembah yang berada diantara jari telunjuk dengan jari tengah. Setelah mendapatkan daftar posisi yang menandakan puncak dan lembah dari diagram distribusi jarak, maka ROI akan didapatkan dari titik P_1 dan P_2 yang dibentuk persegi.

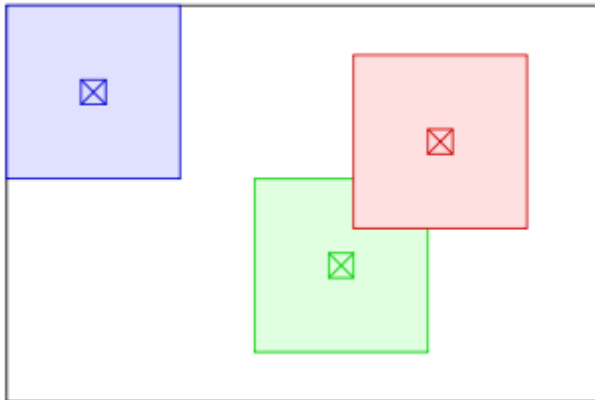
2.6 Adaptive Histogram Equalization

Adaptive Histogram Equalization (AHE) adalah salah satu teknik yang digunakan untuk meningkatkan kontras pada gambar. Metode ini cocok untuk memperbaiki kontras lokal dan meningkatkan definisi tepi pada setiap wilayah gambar. Varian yang ada pada AHE disebut *contrast limited adaptive histogram equalization* (CLAHE). Cara mencegah hal ini salah satunya dengan membatasi amplifikasi.

AHE dapat memperbaiki citra dengan kontras lebih terang atau lebih gelap dengan cara mengubah setiap pixel dengan fungsi transformasi. Dapat disederhanakan bahwa setiap pixel akan ditransformasikan berdasarkan histogram kuadrat yang mengelilingi pixel, seperti pada **Gambar 2.3** [2].

Pixel yang berada pada batas gambar akan diperlakukan khusus karena tidak sepenuhnya masuk ke dalam gambar. Hal tersebut akan diatasi dengan cara memperpanjang gambar dengan memantulkan garis dan kolom pixel yang berkenaan dengan batas gambar. Contoh penggunaan *adaptive histogram equalization*

(*adapthisteq*) seperti pada **Gambar 2.4** untuk citra asli dan **Gambar 2.5** adalah citra setelah *adapthisteq*.



Gambar 2.3 *Adaptive Histogram Equalization*



Gambar 2.4 Citra sebelum dilakukan *adapthisteq* [7]



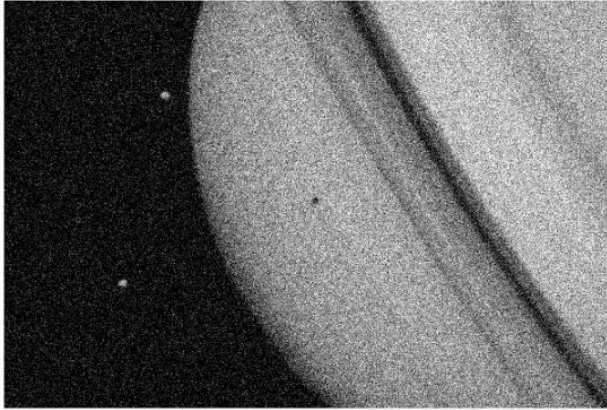
Gambar 2.5 Citra setelah dilakukan *adapthisteq* [7]

2.7 Adaptive Noise Removal

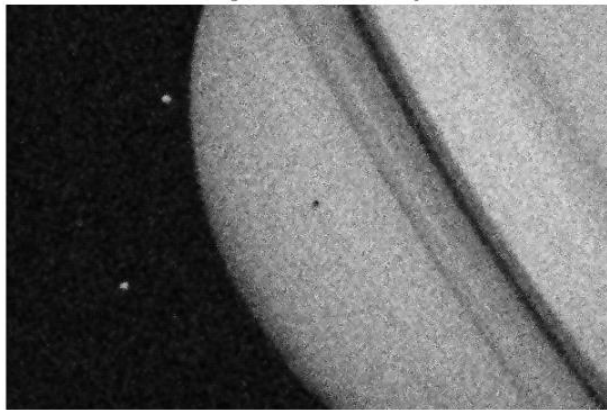
Pengolahan citra menggunakan filter *adaptive noise removal* merupakan salah satu teknik filtering yang digunakan untuk melakukan proses pengurangan *noise* dengan mengganti nilai pixel tetangga dengan nilai baru yang didapat nilai ketetanggaannya. Filter ini bertujuan untuk memperbaiki citra yang awalnya terdapat titik-titik *noise* pada citra menjadi berkurang dan akan terlihat lebih jelas [8]. Algoritma dari teknik filter *adaptive noise removal* ini yaitu:

- Menentukan pixel-pixel mana saja yang akan mempengaruhi perubahan nilai untuk pixel aktual yang selanjutnya akan membentuk graf pixel tetangga.
- Memberi bobot berdasarkan perbedaan jarak antar pixel yang difilter dengan pixel-pixel tetangga. Bobot pixel-pixel tetangga dihitung dengan fungsi *Gaussian*.
- Mengganti nilai pixel dengan rata-rata nilai.
- Filter ini akan memilih pixel tetangga yang relevan. Ukuran dan bentuk pixel tetangga yang akan mempengaruhi perubahan citra.

Perbedaan citra setelah menggunakan filter *adaptive noise removal* ini dapat dilihat pada **Gambar 2.6 dan 2.7**.



Gambar 2.6 Citra Sebelum dilakukan filter Adaptive Noise Removal [9]



Gambar 2.7 Citra Setelah dilakukan filter Adaptive Noise Removal [9]

2.8 Anisotropic Diffusion Filter

Anisotropic Diffusion atau disebut juga dengan *Perona-Malik Diffusio* adalah sebuah teknik yang bertujuan mengurangi noise pada gambar tanpa menghilangkan bagian signifikan dari konten gambar, seperti tepi, garis atau bagian lainnya yang penting

untuk interpretasi gambar. Teknik ini merupakan generalisasi dari proses difusi yang menghasilkan keluarga parameter gambar dengan hasil gambar berupa kombinasi antara gambar asli dan filter yang bergantung pada konten asli gambar [2].

2.9 Frangi Filter

Untuk menonjolkan pola pembuluh darah jari yang ada pada citra, dibutuhkan algoritma *filtering* berupa Frangi Filter. Frangi filter digunakan untuk mencari struktur geometris yang dianggap berbentuk tubular. Deteksi struktur geometris tersebut menggunakan nilai *eigenvalues* dan *Hessian Matrix*.

Hessian Matrix merupakan hasil dari perhitungan turunan fungsi *Gaussian*. Perhitungan untuk mendapatkan *Hessian Matrix* $H_\sigma(x)$ dapat dilihat pada persamaan 2.8,

$$H_\sigma(I, x) = \frac{\partial^2 I_\sigma}{\partial x^2} = I(x) * \frac{\partial^2 \mathcal{G}_\sigma(x)}{\partial x^2} \quad (2.8)$$

dimana I merupakan citra dan \mathcal{G}_σ menyatakan fungsi *Gaussian* dengan standar deviasi σ [4].

Analisis *Hessian Matrix* dilakukan untuk mengekstraksi *eigenvalues* ($|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$) dan *principal direction* ($\bar{u}_1, \bar{u}_2, \bar{u}_3$) dimana struktur *local second order* citra didekomposisi. Hasil analisis tersebut dapat digunakan untuk membedakan pola-pola interpretasi geometris seperti *blob-like*, *plate-like*, tubular maupun bentuk yang tidak beraturan. Karena pembuluh darah jari berbentuk tubular, maka struktur ideal dari pembuluh darah jari akan mengikuti aturan $|\lambda_1| \approx 0, |\lambda_1| \ll |\lambda_2|, \lambda_2 \approx \lambda_3$. Dengan informasi sebelumnya, maka dibentuk fungsi vessel V_F^σ untuk mengukur apakah suatu struktur tertentu mendekati struktur tubular yang ideal. Persamaan fungsi vessel dapat dilihat pada persamaan 2.9,

Dimana parameter α, β, c mengatur tingkat sensitifitas dari proses filtering. Untuk membedakan suatu struktur merupakan sebuah *plate-like* (R_A), *blob-like* (R_B), atau *background* (ϑ) dapat menggunakan persamaan 2.10, 2.11, dan 2.12.

$$V_F^\sigma = \begin{cases} 0, & \text{if } \lambda_2 > 0 \text{ or } \lambda_3 > 0 \\ (1 - \exp\left(-\frac{R_A^2}{2a^2}\right) \exp\left(-\frac{R_B^2}{2\beta^2}\right) (1 - \exp\left(-\frac{\delta^2}{2c^2}\right)), & \end{cases} \quad (2.9)$$

$$R_A = \frac{|\lambda_2|}{|\lambda_3|} \quad (2.10)$$

$$R_A = \frac{|\lambda_1|}{\sqrt{\lambda_2 \lambda_3}} \quad (2.11)$$

$$S = \sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2} \quad (2.12)$$

Bentuk akhir dari persamaan untuk mengukur struktur tubular pembuluh darah dengan Frangi Filter ditunjukkan pada persamaan berikut,

$$V_F = \max_{\sigma_{min} \leq \sigma \leq \sigma_{max}} V_F^\sigma(x) \quad (2.13)$$

2.10 Median Filter

Median Filter merupakan metode penyaringan derau yang dibentuk dari R-Estimator. R-Estimator adalah salah satu *robust estimator* yang penghitungannya didasarkan pada urutan data $x_{(i)}$ dimana $x_{(i)}$ merupakan elemen data pada urutan ke- i dan n menyatakan ukuran himpunan data dengan $1 \leq i \leq n$. Bentuk persamaan dari Median Filter $\hat{\theta}_{mod}$ dapat dimodelkan dalam persamaan 2.14.

$$\hat{\theta}_{mod} = \begin{cases} \frac{1}{2}(X_{(n/2)} + X_{(1+n/2)}), & \text{for even } n \\ X_{(n+1/2)}, & \text{for odd } n \end{cases} \quad (2.14)$$

Dimana jika n merupakan bilangan genap maka nilai Median Filter merupakan setengah dari penjumlahan elemen pada urutan $n/2$ dan elemen pada urutan $1 + n/2$. Sedangkan untuk n merupakan bilangan ganjil maka nilai Median Filter merupakan elemen pada urutan $n + 1/2$. Contoh implementasi median filter bisa dilihat pada gambar 2.8 dan 2.9.



Gambar 2.8 Citra Sebelum dilakukan Median Filter [10]



Gambar 2.9 Citra Setelah dilakukan Median Filter [10]

2.11 Operasi Morfologi

Pada metode operasi morfologi digunakan suatu *structuring element* tertentu yang menjadi cetakan bagi citra keluaran dari hasil analisis citra masukan. Terdapat dua operasi morfologi dasar yaitu

dilasi dan erosi. Kedua operasi tersebut dikenakan pada objek yang terdapat pada citra biner. Objek citra direpresentasikan dengan nilai biner 1 sedangkan background bernilai biner 0.

Pada operasi dilasi wilayah objek pada citra biner akan menjadi lebih luas. Sedangkan pada proses erosi wilayah objek pada citra akan mengerucut menjadi lebih sempit. Jika suatu citra A dikenai *structuring element* B dengan β sebagai anggota dari B maka notasi untuk dilasi dan erosi ditunjukkan masing-masing pada persamaan 2.15 dan 2.16.

$$A \oplus B = \bigcup_{\beta \in B} A + \beta \quad (2.15)$$

$$A \ominus B = \bigcap_{\beta \in B} A - \beta \quad (2.16)$$

Pada operasi dilasi, setiap pixel pada citra A yang tepat berada pada pusat pixel *structuring element* B maka tetangga dari pixel tersebut mengikuti jumlah dan pola dari ketetanggaan pusat pixel *structuring element* B . Sedangkan pada proses operasi erosi, setiap pixel pada citra A yang tepat berada pada pusat pixel *structuring element* B maka tetangga dari pixel tersebut akan dihapus mengikuti jumlah dan pola dari ketetanggaan pusat pixel *structuring element* B .

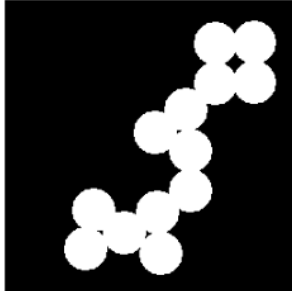
Operasi morfologi dilasi dan rotasi juga dapat saling dikombinasikan membentuk operasi morfologi *closing* dan *opening*. Kedua jenis morfologi ini berfungsi untuk memperhalus kontur objek. Selain itu, operasi *closing* berfungsi menyatukan dua objek yang terpisah, mempertebal area yang tipis, serta mengisi lubang-lubang kecil pada objek. Contoh penerapan *closing* dapat dilihat pada gambar 2.10 dan 2.11.

Fungsi lain dari operasi *opening* yaitu memisahkan dua objek dan menghapus objek-objek kecil. Contoh implementasi fungsi *opening* dapat dilihat pada gambar 2.12 dan 2.13. sedangkan

untuk notasi dari operasi morfologi *opening* dan *closing* dapat diamati pada persamaan berikut,

$$A \circ B = (A \ominus B) \oplus B \quad (2.17)$$

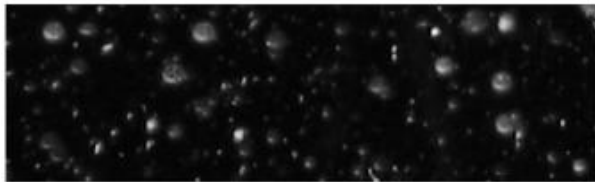
$$A \cdot B = (A \oplus B) \ominus B \quad (2.18)$$



Gambar 2.10 Citra Sebelum dilakukan Morfologi *Closing*
[11]



Gambar 2.11 Citra Setelah dilakukan Morfologi *Closing* [11]



Gambar 2.12 Citra Sebelum dilakukan Morfologi *Opening*
[12]



Gambar 2.13 Citra Setelah dilakukan Morfologi *Opening* [12]

2.12 Similarity

Metode ini digunakan untuk mengukur kemiripan diantara 2 gambar. Pengukuran kualitas gambar didasarkan pada gambar awal yang tidak dikompres. Metode ini dapat memperbaiki metode-metode tradisional seperti Peak Signal-to-Noise Ratio (PSNR) dan Mean Squared Error (MSE) yang tidak sesuai dengan persepsi manusia. Metode ini berbasis model persepsi yang menganggap degradasi gambar sebagai perubahan yang dirasakan pada informasi struktural termasuk pencahayaan luminan dan *masking contrast* [13]. *Similarity index* dihitung pada beberapa jendela gambar. Besar kemiripan diantara 2 jendela x dan y dengan ukuran $N \times N$ dapat dihitung dengan persamaan seperti berikut,

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (2.19)$$

BAB III

ANALISIS DAN PERANCANGAN

Bab analisis dan perancangan berisi analisis kebutuhan dan perancangan aplikasi yang akan dibangun. Tahap analisis membahas mengenai analisis kebutuhan yang menjadi dasar dari tahap perancangan.

3.1 Tahap Analisis

Tahap analisis mendefinisikan kebutuhan yang akan dipenuhi dalam pembangunan aplikasi perbandingan metode ekstraksi fitur menggunakan LLBP dan SIFT pada *palm* dan *finger vein*. Selain itu dijelaskan pula alasan pengerjaan masing-masing tahap pada tugas akhir ini.

3.1.1 Deskripsi Umum

Pada tugas akhir ini dibangun aplikasi untuk melakukan perbandingan metode ekstraksi fitur menggunakan LLBP dan SIFT pada *palm* dan *finger vein*. Data masukan yang digunakan adalah data hasil *preprocessing palm* dan *finger vein*. Data keluaran dari aplikasi merupakan nilai akurasi dari pengenalan yang melalui proses ekstraksi fitur menggunakan LLBP dan SIFT.

Aplikasi ini diharapkan dapat digunakan untuk menunjang dalam hal pengenalan *palm* dan *finger vein* pada bidang keamanan informasi maupun data. Hasil perbandingan dari kedua metode ekstraksi fitur tersebut dapat digunakan sebagai acuan dalam membuat aplikasi pengenalan *palm* dan *finger vein*, agar hasil pengenalan *palm* maupun *finger vein* bisa lebih akurat.

3.1.2 Spesifikasi Kebutuhan Sistem

Pada aplikasi perbandingan metode ekstraksi fitur pada *palm* dan *finger vein* dibutuhkan beberapa proses untuk dapat memenuhi kebutuhan sistem dalam menghasilkan akurasi yang akurat. Proses tersebut antara lain:

3.1.2.1 Pada *Palm Vein*

1. *Preprocessing*

Preprocessing dilakukan untuk memperbaiki citra masukan dan mencari *Region of Interest* (ROI) yang akan menghasilkan citra telapak tangan.

2. Ekstraksi fitur

Ekstraksi fitur dilakukan untuk mendapatkan data yang representatif terhadap bentuk objek dalam citra. Ekstraksi fitur pada *palm vein* menggunakan dua metode, yaitu LLBP dan SIFT.

3. *Similarity*

Similarity dilakukan untuk mencari kesamaan antara citra masukan (data *testing*) dan data training (data yang telah melalui tahap *preprocessing*).

3.1.2.2 Pada *Finger Vein*

1. *Preprocessing*

Preprocessing dilakukan untuk memperbaiki citra masukan dan mencari *Region of Interest* (ROI) akan menghasilkan citra jari yang fokus pada pembuluh darah jari.

2. Ekstraksi fitur

Ekstraksi fitur dilakukan untuk mendapatkan data yang representatif terhadap bentuk objek dalam citra. Ekstraksi fitur pada *palm vein* menggunakan dua metode, yaitu LLBP dan SIFT.

3. *Similarity*

Similarity dilakukan untuk mencari kesamaan antara citra masukan (data *testing*) dan data training (data yang telah melalui tahap *preprocessing*).

3.1.3 Analisis Permasalahan

Hasil perbandingan metode ekstraksi fitur pada *palm* dan *finger vein* dapat dijadikan acuan dalam pembuatan aplikasi pengenalan *palm* maupun *finger vein* agar lebih akurat. Hasil

perbandingan akurasi yang tinggi akan memberikan hasil pengenalan yang lebih akurat pula.

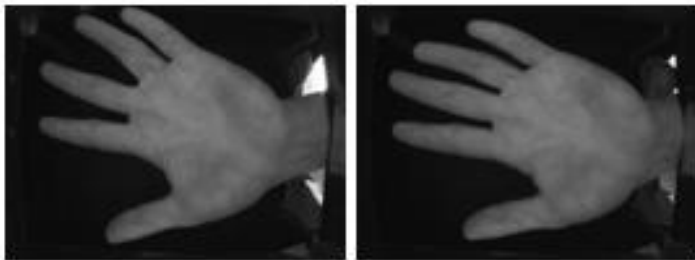
Untuk mencapai hasil pengenalan yang akurat, terdapat beberapa permasalahan yang dapat menurunkan akurasi. Permasalahan-permasalahan tersebut dijelaskan sebagai berikut:

3.1.3.1 Pada Palm Vein

Citra pada pembuluh darah telapak tangan dapat menjadi alternatif untuk melakukan identifikasi pola pada individu memanfaatkan keunikan pola pembuluh darah setiap individu. Akan tetapi citra telapak tangan diambil menggunakan sinar inframerah, maka jaringan yang lain akan mudah terambil. Sebagai contoh bagian jari tangan ikut terambil pada saat pengambilan data telapak tangan. Hal tersebut menyebabkan perlu adanya analisis permasalahan pada citra telapakk tangan.

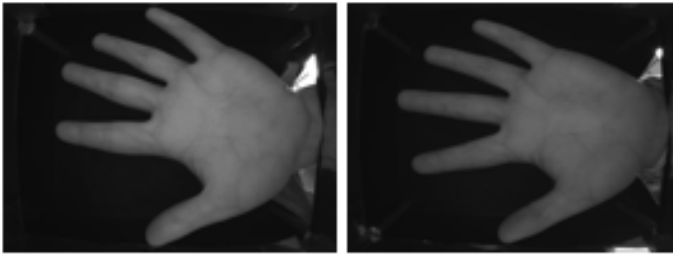
1. Analisis Permasalahan *Preprocessing*

Pada tahap *preprocessing* citra asli dilakukan proses *Region of Interest* (pengambilan citra bagian telapak tangan) dan dilakukan perbaikan citra agar pembuluh darah pada telapak tangan terlihat jelas. Pada proses *region of interest*, terdapat perbedaan rotasi posisi tangan sehingga mempengaruhi proses pemotongan telapak tangan. Tetapi hasil pemotongan tidak berpengaruh pada saat pengenalan telapak tangan. Perbedaan rotasi tangan bisa dilihat pada **Gambar 3.1**.



Gambar 3.1 Perbedaan rotasi tangan

Sedangkan pada proses perbaikan citra, terdapat perbedaan *illumination* atau tingkat kecerahan citra yang mengakibatkan perbedaan pada saat perbaikan citra dan proses selanjutnya. Karena jika intensitas cahaya pada citra asli lebih tinggi. Perbedaan *illumination* bisa dilihat pada **Gambar 3.2**. Pada gambar yang kiri, intensitas cahaya lebih tinggi daripada gambar yang kanan.



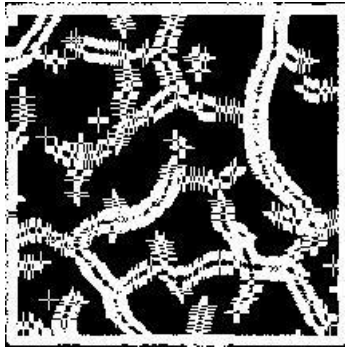
Gambar 3.2 Perbedaan Intensitas Cahaya

2. Analisis Permasalahan Ekstraksi Fitur

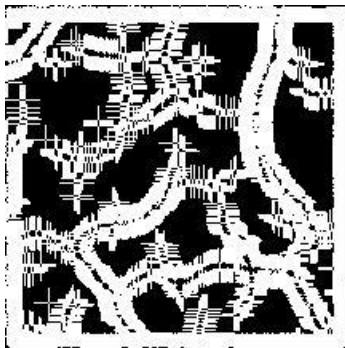
Ekstraksi fitur perlu dilakukan dengan baik, karena data fitur yang sesuai dan representatif setiap citra akan menghasilkan hasil pengenalan yang akurat. Dalam tugas akhir ini, ekstraksi fitur dilakukan dengan menggunakan filter *Local Line Binary Pattern* (LLBP) dan *Scale Invariant Feature Transform* (SIFT).

- Metode Local Line Binary Pattern

Ekstraksi fitur LLBP menggunakan operator pixel-pixel ketetanggaan horizontal dan vertikal terhadap suatu pixel. Pada saat proses memperoleh kode *binary*, panjang pixel operator mempengaruhi hasil ekstraksi fitur. Hasil ekstraksi fitur dengan panjang operator LLBP yang berbeda, bisa dilihat pada **Gambar 3.3** dan **Gambar 3.4**.



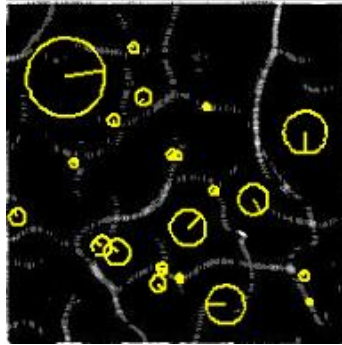
Gambar 3.3 Hasil Ekstraksi Citra *Palm Vein* dengan panjang operator LLBP yaitu 11



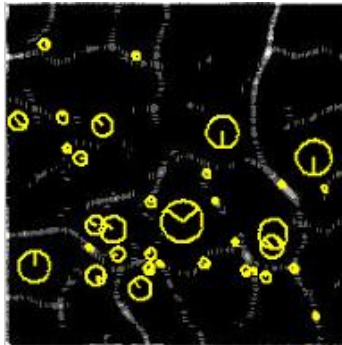
Gambar 3.4 Hasil Ekstraksi Citra *Palm Vein* dengan panjang operator LLBP yaitu 17

- **Metode Scale Invariant Feature Transform**

Pada ekstraksi fitur SIFT, setiap citra akan menghasilkan *keypoint* dan deskriptor yang merepresentasikan citra tersebut. Perbedaan intensitas cahaya pada citra akan mempengaruhi hasil ekstraksi fitur, sehingga *keypoint* yang dihasilkan juga tidak sama dan tidak akurat. Perbedaan *keypoint* pada citra milik individu yang sama dengan perbedaan intensitas cahaya, bisa dilihat pada **Gambar 3.5** dan **Gambar 3.6**.



Gambar 3.5 Hasil *Keypoint* Individu 1 pada Pengambilan Ketiga



Gambar 3.6 Hasil *Keypoint* Individu 1 pada Pengambilan Keempat

3. Analisis Permasalahan *Similarity*

Terdapat perbedaan proses *similarity* pada metode LLBP dan SIFT. Pada metode LLBP menggunakan *structural similarity* sedangkan pada metode SIFT menggunakan *keypoint matching*. Pada *structural similarity*, pola pembuluh darah mempengaruhi proses perhitungan jarak antar citra. Sedangkan pada *keypoint matching* hasil dari penentuan *keypoint* mempengaruhi proses *matching*, karena yang digunakan untuk proses *matching* adalah

deskriptor dari citra tersebut. Sehingga akurasi yang dihasilkan tidak akurat.

3.1.3.2 Pada Finger Vein

Citra pada pembuluh darah jari dapat menjadi alternatif untuk melakukan identifikasi pola pada individu memanfaatkan keunikan pola pembuluh darah setiap individu. Akan tetapi citra jari diambil menggunakan sinar inframerah, maka jaringan yang lain akan mudah terambil sehingga pola pembuluh darah akan ditangkap sebagai bayangan sehingga terlihat samar. Hal tersebut menyebabkan susah untuk membedakan pola jari dengan *background*.

1. Analisis Permasalahan *Preprocessing*

Pada tahap *preprocessing* citra asli jari dilakukan proses perbaikan citra dan *cropping* untuk mendapatkan bagian citra yang fokus pada pembuluh darah jari. Intensitas cahaya citra mengakibatkan perbedaan hasil perbaikan dan proses *cropping* akan mempengaruhi proses ekstraksi fitur dan *matching*, karena pola pembuluh darah jari terlihat samar dengan *background*. Contoh perbedaan *cropping* pada jari dapat dilihat pada **Gambar 3.7** dan **Gambar 3.8**.



Gambar 3.7 *Cropping* diluar *border* jari



Gambar 3.8 *Cropping* didalam *border* jari

2. Analisis Permasalahan Ekstraksi Fitur

Ekstraksi fitur perlu dilakukan dengan baik, karena data fitur yang sesuai dan representatif setiap citra akan menghasilkan hasil pengenalan yang akurat. Dalam tugas akhir ini, ekstraksi fitur dilakukan dengan menggunakan filter *Local Line Binary Pattern* (LLBP) dan *Scale Invariant Feature Transform* (SIFT).

- Metode Local Line Binary Pattern

Ekstraksi fitur LLBP menggunakan operator pixel-pixel ketetangaan horizontal dan vertikal terhadap suatu pixel. Ukuran citra berpengaruh dalam tahap ini, karena pada saat proses memperoleh kode *binary* panjang pixel operator mempengaruhi hasil ekstraksi fitur.

- Metode Scale Invariant Feature Transform

Pada ekstraksi fitur SIFT, setiap citra akan menghasilkan *keypoint* dan deskriptor yang merepresentasikan citra tersebut. Perbedaan intensitas cahaya pada citra akan mempengaruhi hasil ekstraksi fitur, sehingga *keypoint* yang dihasilkan juga tidak sama. Selain itu, hasil dari *preprocessing* juga mempengaruhi, karena perbedaan antara citra pembuluh darah dan *background* terlihat samar.

3. Analisis Permasalahan Similarity

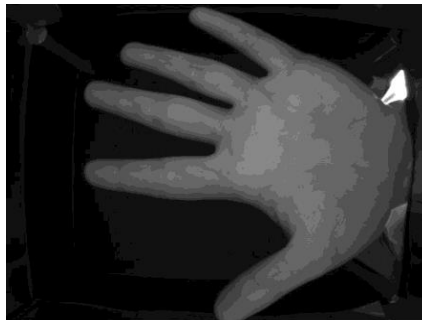
Terdapat perbedaan proses *similarity* pada metode LLBP dan SIFT. Pada metode LLBP menggunakan *structural similarity* sedangkan pada metode SIFT menggunakan *keypoint matching*. Pada *structural similarity* pola pembuluh darah mempengaruhi proses perhitungan jarak antar citra. Sedangkan pada *keypoint matching*, hasil dari penentuan *keypoint* mempengaruhi proses *matching*. Sehingga akurasi yang dihasilkan tidak akurat.

3.2 Tahap Perancangan

Tahap perancangan dilakukan untuk merancang proses secara keseluruhan berdasarkan fungsionalitas dan kebutuhan dari aplikasi perbandingan metode ekstraksi fitur pada *palm* dan *finger vein*.

3.2.1 Perancangan Sistem

Perancangan sistem dilakukan untuk menggambarkan proses secara keseluruhan dari aplikasi perbandingan metode ekstraksi fitur pada *palm* dan *finger vein*. Untuk dapat menggunakan aplikasi perbandingan metode ekstraksi fitur pada *palm* dan *finger vein*, pengguna harus memiliki data masukan. Data masukan diperoleh dari foto *scan palm* dan *finger vein*. Ilustrasi citra *palm* dan *finger vein* ditunjukkan pada **Gambar 3.9** dan **Gambar 3.10**.



Gambar 3.9 Citra *scan palm vein*



Gambar 3.10 Citra *scan finger vein*

Sistem perbandingan metode ekstraksi fitur pada tugas akhir ini memiliki tiga proses utama. Proses pertama adalah *preprocessing*. Proses kedua adalah ekstraksi fitur. Proses ketiga adalah similarity.

Tahap *preprocessing* merupakan tahap pertama yang harus dilakukan untuk perbaikan citra dan pengambilan bagian pembuluh darah citra. Tahap ini terdiri dari proses ROI yang menghasilkan fokus citra pada pembuluh darah telapak tangan dan jari.

Ekstraksi fitur adalah proses untuk mendapatkan nilai representatif dari setiap pixel yang telah melalui proses *preprocessing*. Nilai dari proses ekstraksi fitur tersebut digunakan dalam proses *similarity* dan pembangunan data pada tahap *training* data. Tugas akhir ini menggunakan fitur berupa nilai representatif setiap pixel untuk metode LLBP, yang akan menghasilkan citra ekstraksi fitur. Sedangkan untuk metode SIFT akan menghasilkan nilai *keypoint* dan deskriptor.

Proses *similarity* adalah proses untuk mencari nilai kemiripan antar citra. Proses *similarity* dilakukan berdasarkan pada proses ekstraksi fitur.

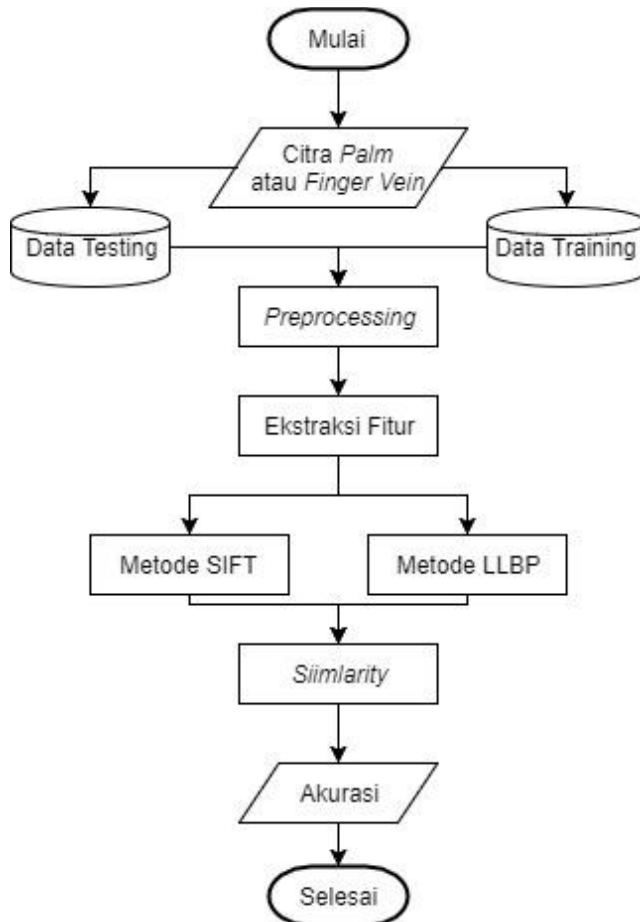
3.2.2 Perancangan Data

Perancangan data dilakukan untuk memastikan pengoperasian aplikasi berjalan dengan benar. Data masukan (*input*) adalah data yang diperlukan dalam pengoperasian aplikasi dan data keluaran (*output*) adalah data yang akan digunakan oleh pengguna.

Data masukan adalah data yang diproses oleh aplikasi perbandingan metode ekstraksi fitur. Data yang digunakan sebagai masukan adalah data hasil *scan palm* dan *finger vein*. Data keluaran pada aplikasi perbandingan metode ekstraksi fitur merupakan nilai akurasi dari hasil *similarity* menggunakan metode LLBP dan SIFT. Data *palm vein* pada tugas akhir ini diambil dari CASIA Multi-Spectral Palmprint Image Data-base V1.0, sedangkan data *finger vein* diperoleh dari *Hongkong Polytechnic University* (Kumar&Zhou, 2012).

3.2.3 Perancangan Proses

Perancangan proses dilakukan untuk memberikan gambaran mengenai setiap proses yang terdapat pada aplikasi perbandingan metode ekstraksi fitur pada *palm* dan *finger vein*. Bagan dari setiap proses utama aplikasi dapat dilihat pada **Gambar 3.11**.



Gambar 3.11 Diagram Alir dari Sistem Perbandingan Metode Ekstraksi Fitur pada Palm Vein dan Finger Vein

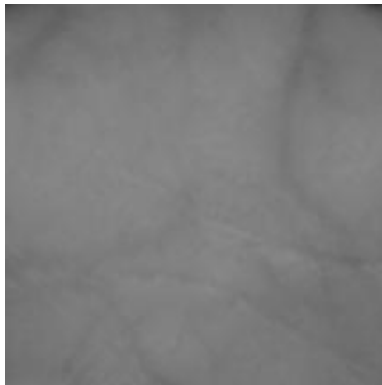
3.2.3.1 *Preprocessing*

Supaya aplikasi dapat berjalan optimal, diperlukan persiapan data terlebih dahulu pada tahap *preprocessing*. Pada tugas akhir kali ini, terdapat dua dataset yang dilakukan *preprocessing* untuk membandingkan kinerja metode ekstraksi fitur pada setiap dataset. Dataset tersebut yaitu data *palm vein* dan *finger vein*.

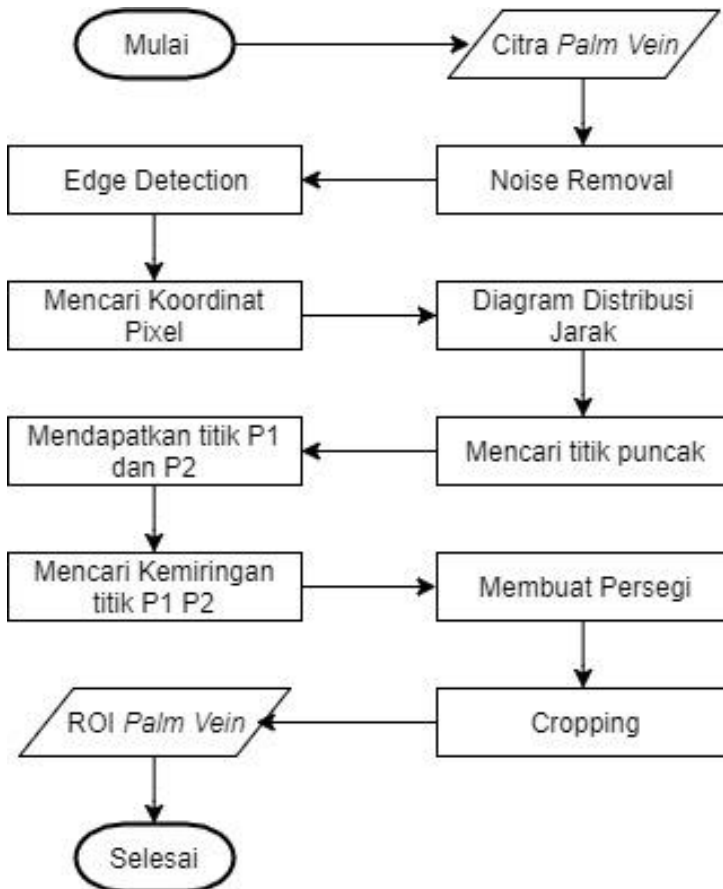
1. Pada Palm Vein

Tahap *preprocessing* yang dilakukan pada *palm vein* adalah pengambilan ROI menjadi bagian utama yaitu bagian terluas dari telapak tangan menggunakan algoritma *inner border tracing*, *resize image* menjadi 216×216 , *median filter*, adaptive histogram equalization, adaptive noise removal, anisotropic diffusion filter, morfologi dengan *image closing*, *subtract image*, dan *adjust intensity*. Diagram alir dari proses ditunjukkan pada **Gambar 3.14**.

Pengambilan ROI dilakukan untuk mendapatkan bagian utama dari telapak tangan, tahap ini menggunakan algoritma *inner border tracing*. Diagram alir proses untuk mendapatkan region of interest dari *palm vein* ditunjukkan pada **Gambar 3.13**. Hasil citra ROI ditunjukkan pada **Gambar 3.12**.



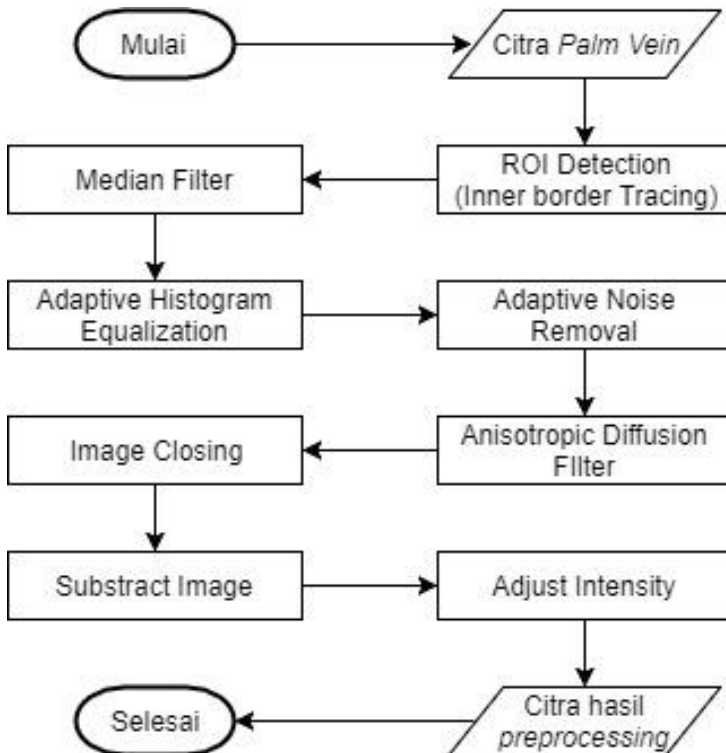
Gambar 3.12 Citra hasil proses *Region of Interest*



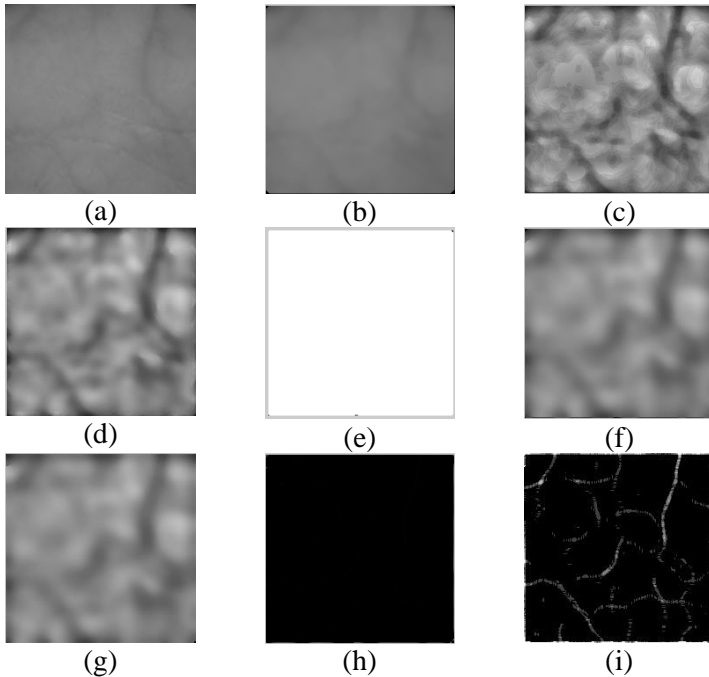
Gambar 3.13 Diagram Alir Proses untuk Mendapatkan Region of Interest dari Palm Vein

Setelah mendapatkan citra ROI, tahap selanjutnya yaitu penghilangan *noise* dan *enhancement* dilakukan dengan menggunakan *median filter*, *adaptive histogram equalization*, dan *adaptive noise removal*. Selanjutnya tahap untuk memisahkan objek pembuluh darah dengan telapak tangan, yaitu dengan

Anisotropic Diffusion Filter, operasi morfologi *closing*, *subtract* citra hasil *Anisotropic Diffusion Filter* dengan citra hasil operasi morfologi *closing*. Selanjutnya tahap untuk memperbaiki intensitas citra atau *adjusting*. Citra *preprocessing* ditunjukkan pada **Gambar 3.15**.



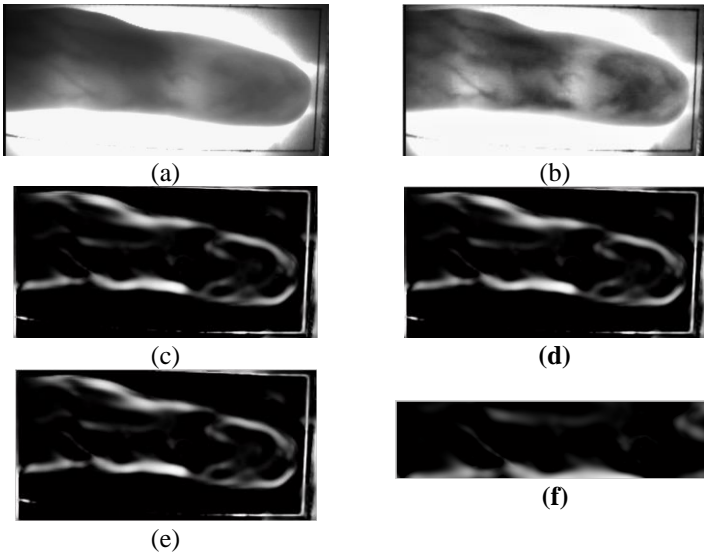
Gambar 3.14 Diagram Alir Tahap *Preprocessing* pada *Palm Vein*



Gambar 3.15 Citra *Preprocessing* (a)Hasil Resize (b)Median Filter (c)Adaptive Histeq (d)Noise Removal (e)ADF (f)Convert (g)Closing (h)Substract (i)Adjust

2. Pada Finger Vein

Tahap *preprocessing* yang dilakukan pada *finger vein* adalah perbaikan citra dan pengambilan ROI menjadi bagian pembuluh darah jari. Perbaikan citra jari ada beberapa tahap yaitu memperjelas *contrast* citra dengan adaptive histogram equalization, frangi filter 2 dimensi untuk menonjolkan pola pada pembuluh darah, *median filter* dan morfologi dengan *image closing* untuk menghilangkan *noise* pada citra, dan dilakukan *cropping* citra untuk mendapatkan bagian pembuluh darah jari. Citra *preprocessing* ditunjukkan pada **Gambar 3.16**. Sedangkan untuk diagram alir dari proses ditunjukkan pada **Gambar 3.17**.

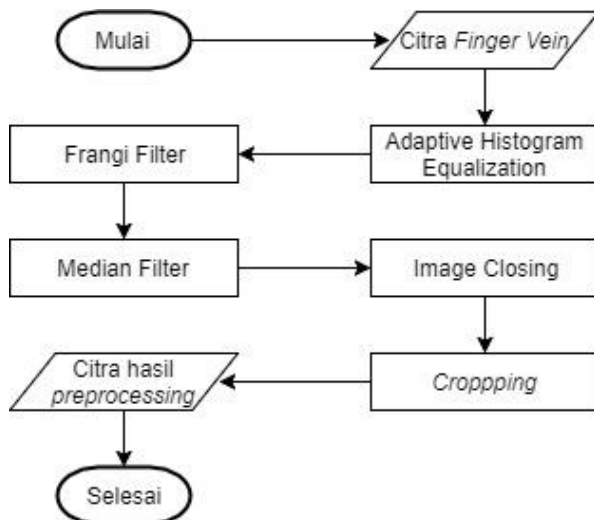


Gambar 3.16 Citra *Preprocessing* (a)Citra Asli (b)Adaptive Histeq (c)Frangi Filter (d)Median Filter (e)*Closing* (f)Hasil *Cropping*

3.2.3.2 Ekstraksi Fitur

Ekstraksi fitur dalam tugas akhir ini menggunakan data keluaran pada tahap *preprocessing*. Pada tahap ekstraksi fitur ini menggunakan metode LLBP dan SIFT. Diagram alir dari proses ekstraksi fitur menggunakan metode LLBP ditunjukkan pada **Gambar 3.18**.

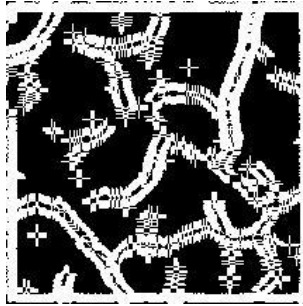
Ekstraksi fitur LLBP dilakukan untuk mendapatkan fitur berupa nilai pixel yang mencirikan pixel citra pembuluh darah. Ekstraksi fitur LLBP dilakukan beberapa kali dengan parameter N (Operator LLBP) yang berbeda, untuk melihat perbandingan nilai hasil ekstraksi fitur LLBP. Setelah mendapatkan operator LLBP akan dilakukan perhitungan kode *binary* horizontal dan vertikal. Setelah itu menghitung nilai *magnitude* dari hasil perhitungan *binary* horizontal dan vertikal. Hasil ekstraksi fitur LLBP ditunjukkan pada **Gambar 3.19**.



Gambar 3.17 Diagram Alir Tahap Preprocessing pada Finger Vein



Gambar 3.18 Diagram Alir Tahap Ekstraksi Fitur LLBP



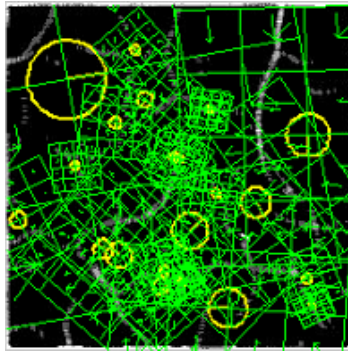
Gambar 3.19 Citra Hasil Ekstraksi Fitur LLBP



Gambar 3.20 Diagram Alir Tahap Ekstraksi Fitur SIFT

Ekstraksi fitur SIFT dilakukan untuk mendapatkan fitur berupa nilai *keypoint* dan deskriptor yang mencirikan suatu citra pembuluh darah. Nilai *keypoint* terdiri dari posisi *keypoint* tersebut (x, y), skala, dan orientasi(dalam radian). Sedangkan untuk

deskriptor merupakan vector dimensi berukuran $128 \times \text{jumlah keypoint}$ dari kelas uint8. Diagram alir tahap ekstraksi fitur SIFT dapat dilihat pada **Gambar 3.20**. Hasil ekstraksi fitur SIFT ditunjukkan pada **Gambar 3.21**, warna kuning menunjukkan *keypoint* pada citra sedangkan warna hijau menunjukkan deskriptor dari *keypoint* yang dihasilkan.



Gambar 3.21 Hasil Ekstraksi Fitur menggunakan metode SIFT

3.2.3.3 *Matching*

Proses *matching* dilakukan untuk pengenalan citra pada data test terhadap data training, sehingga akan menghasilkan citra mana saja yang sesuai dengan data training. Pada setiap data *test* dan data *training* akan dilakukan proses ekstraksi fitur, sehingga setiap citra akan mempunyai *keypoint* dan deskriptor. Setelah mendapatkan *keypoint* dan deskriptor, akan dilakukan proses *matching* yang menggunakan deskriptor sebagai parameternya. Fungsi *matching* akan menemukan deskriptor terdekat antara dua citra menggunakan *euclidean distance*. Indeks kecocokan akan disimpan dalam dua kolom yaitu kecocokan dan jarak antar citra. Proses *matching* ini juga ditambahkan parameter *threshold* untuk menentukan rasio jarak antar *keypoint*.

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

Pada bab ini diuraikan mengenai implementasi perangkat lunak dari rancangan metode yang telah dibahas pada Bab III meliputi kode program dalam perangkat lunak. Selain itu, implementasi dari tiap proses, parameter masukan, keluaran, dan beberapa keterangan yang berhubungan dengan program juga dijelaskan.

4.1 Lingkungan Implementasi

Objek citra yang akan diolah pada implementasi tugas akhir ini adalah sebuah citra *scan palm* dan *finger vein*. Citra *palm vein* adalah gambar tipe jpg, sedangkan citra *finger vein* adalah gambar tipe *bitmap* (bmp).

Dalam implementasi perbandingan metode ekstraksi fitur pada *palm* dan *finger vein* tersebut, digunakan perangkat-perangkat sebagai berikut:

4.1.1 Perangkat Keras

Lingkungan implementasi pada tugas akhir ini adalah sebuah *personal computer* (PC). Perangkat PC yang digunakan adalah tipe desktop *mini tower* bertipe LENOVO BIOS Rev: IEKT24A 0.0 dengan sebuah layar monitor.

Spesifikasi dari PC yang digunakan pada tugas akhir ini adalah: memiliki prosesor Intel Core i3 4150 dengan kecepatan 3,5 GHz dan *Random Access Memory* (RAM) untuk proses menjalankan program sebesar 4,00 GB.

4.1.2 Perangkat Lunak

Lingkungan implementasi pada tugas akhir ini adalah sebuah *personal computer* (PC). Spesifikasi PC dari sisi perangkat lunak menggunakan *software* MATLAB R2014a. Penggunaan MATLAB didukung dengan dua *toolbox* utama yaitu *statistical and machine learning toolbox* dan *image processing toolbox*.

Selain itu, pada tugas akhir ini dalam melakukan pengolahan angka didukung dengan *software Microsoft Excel*. Pembuatan diagram alir proses pada tugas akhir ini menggunakan draw.io.

4.2 Implementasi Tahap *Preprocessing*

Tahap *preprocessing* pada tugas akhir ini diimplementasikan pada dua dataset yaitu pada dataset *palm* dan *finger vein*. Penjelasan dari masing-masing dataset tersebut adalah sebagai berikut:

4.2.1 Pada Palm Vein

Tahap *preprocessing* pada *palm vein* terdiri dari dua tahap. Tahap pertama adalah *ROI detection* dan perbaikan citra untuk mendapatkan pola pembuluh darah telapak tangan. Penjelasan dari masing masing tahapan tersebut adalah sebagai berikut:

4.2.1.1 Implementasi *ROI Detection*

Tahap ini mengambil bagian telapak tangan dengan menggunakan algoritma *inner border tracing*. Implementasi *ROI detection* ditunjukkan pada kode sumber 4.1. Sebelum dilakukan algoritma *inner border tracing*, citra dilakukan proses *opening* yang bertujuan untuk menghilangkan *noise* yang terdapat pada citra *palm vein*. Proses *opening* dapat dilihat pada baris 1 dan 2. Pada baris 4 sampai 7 untuk menentukan mencari nilai koordinat pixel objek yang pertama ditemukan. Pada baris 13 inisialisasi *dir* dengan nilai 7, karena pada tugas akhir ini menggunakan 8-*connectivity*. Pada baris 16 sampai 20 digunakan untuk mendapatkan nilai arah dari pixel tetangga, jika genap menggunakan rumusan $(\text{posisi pixel} + 7) \bmod 8$, sedangkan untuk ganjil menggunakan $(\text{posisi pixel} + 6) \bmod 8$. Baris 23 sampai 74 berfungsi untuk menentukan posisi dari pixel tetangga berdasarkan nilai *dir* yang didapatkan dan pencaarian nilai arah dengan menggunakan 8-*connectivity* serta berlawanan dengan jarum jam.

Pada baris 77 sampai 83 berfungsi untuk mengatur bentuk diagram, dimana koordinat pertama pada diagram merupakan koordinat pixel yang memiliki jarak paling kecil. Diagram disini yang dimaksud yaitu diagram distribusi jarak yang digunakan untuk pengambilan titik P_1 dan P_2 , dimana P_1 adalah titik lembah yang berada diantara jari kelingking dan jari manis, sedangkan P_2 adalah titik lembah yang berada diantara jari telunjuk dan jari tengah. Pada baris 83 sampai 86 berfungsi untuk menghilangkan *noise* dengan cara melakukan *thresholding* dimana pixel yang mempunyai posisi sumbu y diatas 550 akan memiliki jarak bernilai 0. Baris 87 berfungsi untuk memanggil fungsi *fpeak*, kode sumber fungsi *fpeak* dapat dilihat pada kode sumber 4.2.

Pada baris 88 sampai 95 berfungsi untuk menghapus duplikasi nilai puncak setelah mendapatkan titik P_1 dan P_2 . Pada baris 106 sampai 109 berfungsi untuk mengetahui kemiringan antara titik P_1 dan P_2 yang selanjutnya digunakan untuk membentuk persegi. Pada baris 110 sampai 118 berfungsi untuk mendapatkan titik P_1 dan P_2 . Pada baris 119 sampai 121 berfungsi untuk mendapatkan titik P_3 dan P_4 . Pada baris 122 berfungsi untuk melakukan *cropping* dari titik yang telah ditentukan menjadi persegi, sedangkan untuk baris 123 berfungsi untuk mengganti ukuran citra hasil *cropping* menjadi 150×150 .

1	<code>se = strel('disk',17);</code>
2	<code>imgbw = imopen(imgbw_in,se);</code>
	<code>ctrImg =</code>
	<code>zeros(size(imgbw,1),s</code>
3	<code>ize(imgbw,2));</code>
4	<code>[r c] = find(imgbw==1,1);</code>
5	<code>ctrImg(r,c) = 1;</code>
6	<code>coor = [r,c];</code>
7	<code>startCoor = [r,c];</code>
8	<code>maksCoorR = size(img,1);</code>
9	<code>maksCoorC = size(img,2);</code>
10	<code>coorHalfWrist(1,1) = maksCoorR;</code>
11	<code>coorHalfWrist(1,2) = maksCoorC/2;</code>
12	<code>count = 1;</code>

13	dir = 7;
14	indikator = 1;
15	while indikator == 1
16	if mod(dir,2)==0
17	startDir = mod(dir+7,8);
18	else
19	startDir = mod(dir+6,8);
20	end
21	currentDir = startDir;
22	for i=1:8
23	if currentDir==8
24	currentDir = 0;
25	end
26	switch (currentDir)
27	case 0
28	posisi = [0,1];
29	dir = 0;
30	case 1
31	posisi = [-1,1];
32	dir = 1;
33	case 2
34	posisi = [-1,0];
35	dir = 2;
36	case 3
37	posisi = [-1,-1];
38	dir = 3;
39	case 4
40	posisi = [0,-1];
41	dir = 4;
42	case 5
43	posisi = [1,-1];
44	dir = 5;
45	case 6
46	posisi = [1,0];
47	dir = 6;
48	case 7
49	posisi = [1,1];
50	dir = 7;

51	end
52	newCoor = coor + posisi;
53	if newCoor(1,1) > 0 && newCoor(1,2) > 0 && newCoor(1,1) < maksCoorR && newCoor(1,2) < maksCoorC
54	if imgbw(newCoor(1,1),newCoor(1,2))== 1
55	ctrImg(newCoor(1,1),newCoor(1,2))=1;
56	coor = newCoor;
57	a = (newCoor(1,1)- coorHalfWrist(1,1))^2;
58	b = (newCoor(1,2)- coorHalfWrist(1,2))^2;
59	distance = sqrt(a + b);
60	coorDis(count,1) = newCoor(1,1);
61	coorDis(count,2) = newCoor(1,2);
62	coorDis(count,3) = distance;
63	direction(count,1) = dir;
64	count= count + 1;
65	break;
66	else
67	currentDir = currentDir + 1;
68	end
69	else
70	currentDir = currentDir + 1;
71	end
72	end
73	if startCoor == coor
74	indikator = 0;
75	end
76	end
77	for i = 1:size(coorDis(:,3))
78	sumbuX(1,i) = i;

79	<code>end</code>
80	<code>[r c] = find(coorDis(:,3) == min(coorDis(:,3)));</code>
81	<code>newCoorDis = zeros(size(coorDis,1),size(coorDis,2),size(coorDis,3));</code>
82	<code>newCoorDis(1:size(coorDis,1)-r+1,:) = coorDis(r:size(coorDis,1),:);</code>
83	<code>newCoorDis(size(coorDis,1)-r+2:size(coorDis,1),:) = coorDis(1:r-1,:);</code>
84	<code>zeroDis = 550;</code>
85	<code>[r c] = find(newCoorDis(:,1)>zeroDis);</code>
86	<code>newCoorDis(r,3) = 0;</code>
87	<code>output = fpeak(sumbuX,newCoorDis(:,3),50);</code>
88	<code>for i=1:size(output,1)-1</code>
89	<code> if output(i,2)==output(i+1,2)</code>
90	<code> output(i,3) = 1;</code>
91	<code> else</code>
92	<code> output(i,3) = 0;</code>
93	<code> end</code>
94	<code>end</code>
95	<code>improveOut = find(output(:,3)==1);</code>
96	<code>improveOut = sort(improveOut,'descend');</code>
97	<code>for i=1:size(improveOut)</code>
98	<code> output(improveOut(i,1),:)=[];</code>
99	<code>end</code>
100	<code>pixelP1 = output(3,1);</code>
101	<code>pixelP2 = output(7,1);</code>
102	<code>coorP1(1,1) = newCoorDis(pixelP1,1);</code>
103	<code>coorP1(1,2) = newCoorDis(pixelP1,2);</code>
104	<code>coorP2(1,1) = newCoorDis(pixelP2,1);</code>
105	<code>coorP2(1,2) = newCoorDis(pixelP2,2);</code>
106	<code>temp = (coorP2(1,2)-coorP1(1,2))/(coorP2(1,1)-coorP1(1,1));</code>
107	<code>rad = atan(temp);</code>
108	<code>deg = rad/pi*180;</code>

	<code>newImg = imrotate(img, 90-</code>
109	<code>deg, 'bilinear', 'crop');</code>
110	<code>temp = zeros(size(img,1), size(img,2));</code>
111	<code>temp(coorP1(1,1), coorP1(1,2)) = 1;</code>
112	<code>temp(coorP2(1,1), coorP2(1,2)) = 1;</code>
	<code>tempRotate = imrotate(temp, 90-</code>
113	<code>deg, 'bilinear', 'crop');</code>
114	<code>[i, j] = find(tempRotate);</code>
115	<code>newCoorP1(1,1) = i(1);</code>
116	<code>newCoorP1(1,2) = j(1);</code>
117	<code>newCoorP2(1,1) = i(size(i,1));</code>
118	<code>newCoorP2(1,2) = j(size(j,1));</code>
119	<code>a = (newCoorP1(1,1)-newCoorP2(1,1))^2;</code>
120	<code>b = (newCoorP1(1,2)-newCoorP2(1,2))^2;</code>
121	<code>distance = sqrt(a + b);</code>
	<code>fiturI = imcrop(newImg, [newCoorP1(1,2),</code>
122	<code>newCoorP1(1,1), distance , distance]);</code>
123	<code>fiturImg = imresize(fiturI, [150 150]);</code>

Kode Sumber 4.1 Implementasi ROI Detection

4.2.1.2 Implementasi Pencarian *fpeak* pada proses ROI Detection

Proses pencarian *fpeak* ini berfungsi untuk mencari puncak dari setiap jari, kemudian akan menghasilkan keluaran yang digunakan untuk menentukan titik P_1 dan P_2 .

1	<code>Data = sortrows([x,y]);</code>
2	<code>for i=1:rx</code>
3	<code> if i-s<1</code>
4	<code> top = 1;</code>
5	<code> else</code>
6	<code> top = i-s;</code>
7	<code> end</code>
8	<code> y = Data(:,2);</code>
9	<code> if i+s>length(y)</code>
10	<code> bottom = length(y);</code>
11	<code> else</code>
12	<code> bottom = i + s;</code>

13	<code>end</code>
14	<code>if sum(isnan(isP))==0</code>
15	<code>output(numP,:) = isP; %#ok</code>
16	<code>numP = numP + 1;</code>
17	<code>end</code>
18	<code>end</code>

Kode Sumber 4.2 Implementasi Pencarian *fpeak* pada Proses ROI Detection

4.2.1.3 Implementasi Perbaikan Citra

Perbaikan citra dilakukan untuk mendapatkan pola pembuluh darah pada telapak tangan. Sebelum mendapatkan pola pembuluh darah, dilakukan proses perbaikan citra terlebih dahulu yang ditunjukkan pada baris 1 sampai 3. Sedangkan unntuk mendapatkan pola pembuluh darah dapat dilihat pada baris 4 sampai 8. Pada baris 4 bertujuan untuk menonjolkan pola pembuluh darah, baris 5 untuk merubah nilai double menjadi tipe gambar, baris 8 melakukan pengurangan citra hasil penonolan pembuluh darah dengan hasil *closing*, sedangkan baris 9 bertujuan untuk memperbaiki intensitas citra hasil pengurangan.

1	<code>immed = medfilt2(b,[10 10]);</code>
	<code>imadapt = adapthisteq(immed,'clipLimit',0.08,'Distribution','rayleigh');</code>
2	
	<code>imanr = wiener2(imadapt,[12 12]);</code>
3	
4	<code>imadf = anisodiff2D(imanr, 50, 1/7, 20, 1);</code>
5	<code>imapa= uint8(round(imadf-1));</code>
6	<code>se = strel('disk',3);</code>
7	<code>imageClose = imclose(imapa,se);</code>
8	<code>imsub = imsubtract(imageClose,imapa);</code>
9	<code>imadj = imadjust(imsub);</code>

Kode Sumber 4.3 Implementasi Perbaikan Citra pada *Palm Vein*

4.2.2 Pada Finger Vein

Tahap *preprocessing* pada *finger vein* terdiri dari dua tahap. Tahap pertama adalah perbaikan citra untuk mendapatkan pola pembuluh darah serta memotong bagian pembuluh darah jari (*cropping*), dan mendapatkan pola pembuluh darah telapak tangan. Penjelasan dari masing masing tahapan tersebut adalah sebagai berikut:

4.2.2.1 Implementasi Perbaikan Citra

Pada tahap perbaikan citra dilakukan untuk memperoleh pola dari pembuluh darah pada jari. Pada baris 1 dilakukan untuk memperjelas *contrast* dari citra jari. Pada baris 2 dilakukan perubahan tipe matriks yang dihasilkan dari baris 1, tipe data pada matriks dirubah menjadi double karena pada baris 3 akan dilakukan fungsi *frangifilter* 2 dimensi yang inputnya harus dalam bentuk double. Pada baris 3 dilakukan proses *frangifilter* 2 dimensi untuk menonjolkan pola pada pembuluh darah, fungsi ini menggunakan 2 dimensi karena cocok digunakan jika citra masukannya adalah hasil *scan*. Pada baris 4 sampai 6 dilakukan untuk menghilangkan *noise* pada citra.

1	<code>imAdapt = adapthisteq(img, 'clipLimit', 0.01);</code>
2	<code>imDouble = double(imAdapt);</code>
3	<code>imff = FrangiFilter2D(imDouble);</code>
4	<code>imMedFilt = medfilt2(imff);</code>
5	<code>se = strel('disk', 2);</code>
6	<code>imgClose = imclose(imMedFilt, se);</code>

Kode Sumber 4.4 Implementasi Perbaikan Citra pada *Finger Vein*

4.2.2.2 Implementasi Pemotongan pada Bagian Pembuluh Darah

Pemotongan pada bagian pembuluh darah jari dilakukan secara manual menggunakan *tools* Matlab seperti pada kode

sumber x baris 1. Pada fungsi `imcrop` terdapat dua parameter yaitu citra yang akan dilakukan pemotongan dan ukuran yang akan dipotong. Pemotongan ini berbentuk *rectangle* dengan ukuran x minimal 7, y minimal 79, lebar 360, dan tinggi 86.

1	<code>[imgc, rect] = imcrop(imgClose,[7 79 360 86]);</code>
---	---

Kode Sumber 4.5 Implementasi Pemotongan pada Bagian Pembuluh Darah

4.3 Implementasi Ekstraksi Fitur

Tahap ekstraksi fitur pada tugas akhir ini menggunakan dua metode yaitu LLBP dan SIFT. Penjelasan dari masing-masing metode tersebut adalah sebagai berikut:

4.3.1 Implementasi Ekstraksi Fitur LLBP

Pada tahap ekstraksi fitur menggunakan metode LLBP ini terdapat 5 tahap yaitu mengekstraksi kode biner sekaligus menghitung total pembobotan kode biner pada sisi kiri, sisi kanan, sisi atas, dan sisi bawah pixel utama. Sisi kiri dan kanan merupakan LLBP pada arah horizontal, sedangkan pada sisi atas dan bawah merupakan LLBP arah vertikal. Proses menghitung nilai-nilai tersebut akan dilakukan secara berulang sesuai dengan jumlah pixel pada citra.

4.3.1.1 Implementasi LLBP pada Sisi Kiri

Pada proses implementasi metode LLBP pada sisi kiri ini merupakan bagian perhitungan nilai LLBP horizontal. Pada baris 1 akan menyimpan panjang maksimal pixel, sehingga perulangan untuk ekstraksi kode biner akan dilakukan dari indeks 1 hingga batas maksimal paling kiri. Variabel `c_llbp` diperoleh dari panjang pixel (N) dibagi dengan 2, karena N menunjukkan panjang pixel horizontal yang akan digunakan untuk mencari nilai LLBP horizontal. Variabel `dataxy` menyimpan nilai pixel utama, sedangkan untuk variabel kiri akan diinisialisasi dengan nilai 0. Pada baris 3 sampai 6 berfungsi untuk mengekstraksi kode biner dan menghitung sebagian nilai LLBP horizontal.

1	<code>k = max(1,i-c_llbp);</code>
2	<code>while k <= i</code>
3	<code> kiri = kiri * 2;</code>
4	<code> data = imgTest(k,j);</code>
5	<code> if(data > dataxy)</code>
6	<code> kiri = kiri + 1;</code>
7	<code> end</code>
8	<code> k = k + 1;</code>
9	<code>end</code>

Kode Sumber 4.6 Implementasi LLBP pada Sisi Kiri

4.3.1.2 Implementasi LLBP pada Sisi Kanan

Pada proses implementasi metode LLBP pada sisi kanan ini merupakan bagian perhitungan nilai LLBP horizontal. Pada baris 1 akan menyimpan panjang maksimal pixel, sehingga perulangan untuk ekstraksi kode biner akan dilakukan dari indeks sebelah kanan setelah pixel utama hingga batas maksimal paling kanan. Untuk variabel kanan akan diinisialisasi dengan nilai 0. Pada baris 3 sampai 6 berfungsi untuk mengekstraksi kode biner dan menghitung sebagian nilai LLBP horizontal.

1	<code>k = min(valx,i+c_llbp);</code>
2	<code>while k >= i</code>
3	<code> kanan = kanan * 2;</code>
4	<code> data = imgTest(k,j);</code>
5	<code> if(data > dataxy)</code>
6	<code> kanan=kanan+1;</code>
7	<code> end</code>
8	<code> k = k - 1;</code>
9	<code>end</code>

Kode Sumber 4.7 Implementasi LLBP pada Sisi Kanan

4.3.1.3 Implementasi LLBP pada Sisi Atas

Pada proses implementasi metode LLBP pada sisi atas ini merupakan bagian perhitungan nilai LLBP vertikal. Pada baris 1 akan menyimpan panjang maksimal pixel, sehingga perulangan untuk ekstraksi kode biner akan dilakukan dari indeks sebelah atas

setelah pixel utama hingga batas maksimal adalah paling atas (terhitung $N/2$ setelah pixel utama). Untuk variabel atas akan diinisialisasi dengan nilai 0. Pada baris 3 sampai 6 berfungsi untuk mengekstraksi kode biner dan menghitung sebagian nilai LLBP vertikal.

1	<code>k = max(1, j - c_llbp);</code>
2	<code>while k <= j</code>
3	<code> atas = atas * 2;</code>
4	<code> data = imgTest(i, k);</code>
5	<code> if (data > dataxy)</code>
6	<code> atas = atas + 1;</code>
7	<code> end</code>
8	<code> k = k + 1;</code>
9	<code>end</code>

Kode Sumber 4.8 Implementasi LLBP pada Sisi Atas

4.3.1.4 Implementasi LLBP pada Sisi Bawah

Pada proses implementasi metode LLBP pada sisi bawah ini merupakan bagian perhitungan nilai LLBP vertikal. Pada baris 1 akan menyimpan panjang maksimal pixel, sehingga perulangan untuk ekstraksi kode biner akan dilakukan dari indeks sebelah bawah setelah pixel utama hingga batas maksimal adalah paling bawah (terhitung $N/2$ setelah pixel utama). Untuk variabel bawah akan diinisialisasi dengan nilai 0. Pada baris 3 sampai 6 berfungsi untuk mengekstraksi kode biner dan menghitung sebagian nilai LLBP vertikal.

1	<code>k = min(valy, j + c_llbp);</code>
2	<code>while k >= j</code>
3	<code> bawah = bawah * 2;</code>
4	<code> data = imgTest(i, k);</code>
5	<code> if (data > dataxy)</code>
6	<code> bawah = bawah + 1;</code>
7	<code> end</code>
8	<code> k = k - 1;</code>
9	<code>end</code>

Kode Sumber 4.9 Implementasi LLBP pada Sisi Bawah

4.3.1.5 Implementasi Perhitungan *Magnitude* LLBP

Pada tahap ini dilakukan perhitungan nilai LLBP setiap pixel. Nilai tersebut nantinya akan merepresentasikan nilai setiap pixel. Perhitungan ini diperoleh dari hasil LLBP horizontal dan vertikal. LLBP horizontal sendiri diperoleh dari hasil LLBP pada sisi kiri dan kanan dipangkat 2, sedangkan untuk LLBP vertikal diperoleh dari hasil LLBP pada sisi atas dan kanan dipangkat 2. Hasil dari LLBP horizontal dipangkat 2 dijumlahkan dengan LLBP vertikal dipangkat 2, kemudian diakar pangkat 2. Hasil dari nilai tersebut merupakan nilai representatif atau bisa disebut dengan nilai fitur dari pixel tersebut.

1	<code>dist(i, j) = sqrt((kiri+kanan)^2 + (atas+bawah)^2);</code>
---	--

Kode Sumber 4.10 Implementasi Perhitungan *Magnitude* LLBP

4.3.2 Implementasi Ekstraksi Fitur SIFT

Implementasi ekstraksi fitur menggunakan metode SIFT ini memanfaatkan library dari Matlab sesuai dengan referensi yang digunakan yaitu metode SIFT berdasarkan D.Lowe. Ekstraksi fitur ini akan mendapatkan nilai *frame* (*keypoint*) dan *descriptor* yang mencirikan suatu citra. Pada variable *f* terdiri dari [x, y, s, th]. *x* dan *y* menunjukkan posisi setiap *keypoint*, *s* adalah skala dari *keypoint* tersebut, sedangkan *th* menunjukkan orientasi atau arah dari setiap *keypoint* (dalam radian).

1	<code>[f, d] = vl_sift(I);</code>
---	-----------------------------------

Kode Sumber 4.11 Implementasi Ekstraksi Fitur SIFT

4.3.3 Implementasi *Similarity* pada *Palm Vein*

Pada tahap *similarity* pada citra hasil ekstraksi fitur pembuluh darah menggunakan metode LLBP ini memanfaatkan library dari Matlab yaitu menggunakan *ssim* (*Structural Similarity Index*). Fungsi ini melakukan *similarity* berdasarkan pencahayaan, kontras, dan struktur citra. Perhitungan proses *similarity* menggunakan mean, standard deviasi, dan covarians.

1	<code>nilaisim(d, j) = ssim(img1, img2)</code>
---	--

Kode Sumber 4.3 Implementasi *Similarity* pada metode LLBP

4.3.4 Implementasi Matching pada Finger Vein

Pada tahap *matching* pada citra hasil ekstraksi fitur pembuluh darah menggunakan metode SIFT ini memanfaatkan library dari Matlab yaitu `vl_ubcmatch`. Fungsi ini melakukan *matching* berdasarkan jarak antara dua citra yang memanfaatkan *L2 norm*. Fungsi ini mencari deskriptor yang sama antar citra, sehingga demikian deskriptor setiap citra digunakan sebagai parameter untuk melakukan *matching*. Selain deskriptor, *threshold* bisa digunakan sebagai parameter untuk *matching*. Hasil dari proses ini yaitu *matches* dan *scores*. Variabel *matches* menyimpan indeks *keypoint* yang sama antar kedua citra, sedangkan *scores* menyimpan jarak antar *keypoint* yang sama [14].

1	<code>[matches, scores] = vl_ubcmatch(d1, d2, threshold) ;</code>
---	---

Kode Sumber 4.4 Implementasi *Matching* pada Metode SIFT

BAB V

UJI COBA DAN EVALUASI

Dalam bab ini dibahas mengenai hasil uji coba sistem yang telah dirancang dan dibuat. Uji coba dilakukan untuk mengetahui kinerja sistem dengan lingkungan uji coba yang telah ditentukan.

5.1 Lingkungan Uji Coba

Lingkungan uji coba pada tugas akhir ini adalah sebuah *personal computer* (PC). Spesifikasi PC dari sisi perangkat keras adalah memiliki prosesor Intel Core i3 4150 dengan kecepatan 3,5 GHz dan memori untuk proses sebesar 4,00 GB. PC yang digunakan memiliki sistem operasi Windows 8.

Pada sisi perangkat lunak, uji coba pada tugas akhir ini dilakukan dengan menggunakan *software* MATLAB R2014a. Penggunaan MATLAB didukung dengan dua *toolbox* utama yaitu *statistical and machine learning toolbox* dan *image processing toolbox*. Dokumentasi hasil uji coba dilakukan dengan menggunakan *Microsoft Paint*.

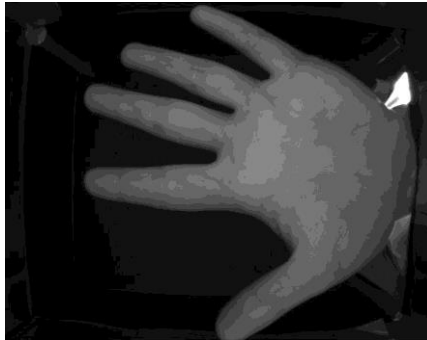
5.2 Data Uji Coba

Data uji coba yang digunakan sebagai masukan adalah citra hasil *scan palm* dan *finger vein*. Contoh citra hasil *scan palm* dan *finger vein* ditunjukkan pada **Gambar 5.1 dan 5.2**. Dari citra *palm vein* dilakukan proses *ROI detection* untuk mendapatkan bagian telapak tangan yang terdapat pembuluh darah. Ukuran ROI yang digunakan adalah 216×216 pixel. Contoh hasil dari proses *cropping* ditunjukkan pada **Gambar 5.3 dan 5.4**.

Untuk menguji performa dua metode ekstraksi fitur akan dilakukan proses *recognition* 35 data test dan 175 data training dari *palm vein* dan *finger vein*. Pemilihan data test dan data training berdasarkan nomor urut pengambilan citra *scan*. Setiap citra *finger* maupun *palm vein* dilakukan pengambilan sebanyak 6 kali, untuk data testing pada *palm* maupun *finger vein* diambil dari citra pengambilan pertama. Sedangkan untuk data training 5 data selain

data testing. Proses *recognition* dilakukan dengan cara *similarity* dan *matching* antara data training dan data testing, sehingga akan mendapatkan akurasi antar metode yang diimplementasikan pada dataset *palm* maupun *finger vein*. Sehingga akan menghasilkan 4 akurasi pembandingan.

Dalam pengujian ini, kualitas data sangat mempengaruhi proses *recognition*. Sehingga pada uji coba ini juga akan menggunakan 35 data testing *palm* maupun *finger vein*.



Gambar 5.1 Citra Hasil Scan Palm Vein



Gambar 5.2 Citra Hasil Proses ROI Detection



Gambar 5.3 Citra Hasil Scan Finger Vein



Gambar 5.4 Citra Hasil Proses Cropping

5.3 Skenario Uji Coba

Uji coba dilakukan untuk mengetahui nilai-nilai parameter yang tepat untuk digunakan pada masing-masing proses. Nilai parameter yang tepat penting untuk diketahui karena penggunaan parameter yang tepat akan memberikan hasil yang terbaik pada keluaran tiap proses.

Skenario pengujian terdiri dari empat macam yaitu:

1. Uji coba mendapatkan akurasi pada *palm vein* menggunakan metode ekstraksi fitur LLBP dengan panjang operator LLBP tertentu.
2. Uji coba mendapatkan akurasi pada *finger vein* menggunakan metode ekstraksi fitur LLBP dengan panjang operator LLBP tertentu.
3. Uji coba mendapatkan akurasi pada *palm vein* menggunakan metode ekstraksi fitur SIFT dengan *threshold* tertentu.
4. Uji coba mendapatkan akurasi pada *finger vein* menggunakan metode ekstraksi fitur SIFT dengan *threshold* tertentu.

5.4 Uji Coba Mendapatkan Akurasi pada *Palm Vein Recognition* menggunakan Metode Ekstraksi Fitur LLBP

Uji coba pengenalan pada telapak tangan menggunakan deteksi pola pembuluh darah ini akan menghasilkan akurasi untuk setiap panjang operator LLBP. Tujuan dari uji coba ini yaitu untuk mengetahui performa metode LLBP pada proses ekstraksi fitur jika digunakan *recognition* pada setiap individu dengan perbedaan jumlah dan kualitas data testing. Kualitas citra yang bagus dilihat dari hasil *preprocessing* yang menunjukkan pola pembuluh darah lebih jelas.

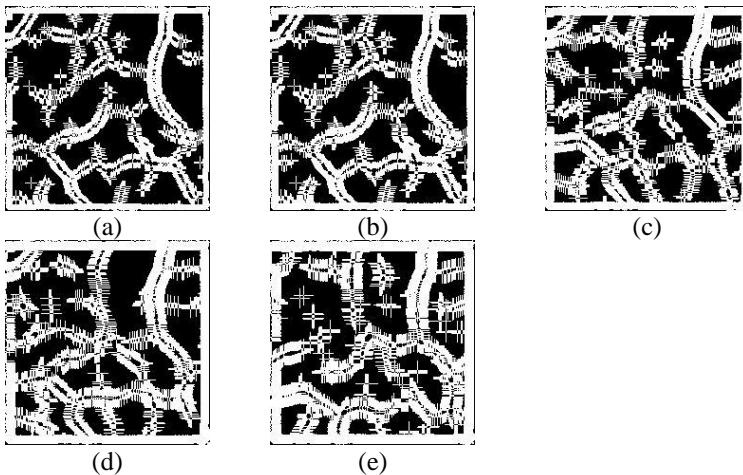
Pengujian dilakukan dengan mendapatkan nilai akurasi pada setiap panjang operator LLBP. Pengujian dilakukan dengan menggunakan fungsi *similarity* pada MATLAB. Jika hasil *similarity* tinggi dan data testing sesuai dengan data training maka akan menambah nilai akurasi. Nilai panjang operator yang digunakan dalam uji coba adalah 11, 13, 15, 17, dan 19.

Tabel 5.1 Hasil Uji Coba Metode LLBP pada *Palm Vein*

PALM VEIN LLBP	
Akurasi	
N	Data Testing
	35
11	88.60%
13	97.14%
15	94.29%
17	85.71%
19	62.86%

Hasil uji coba ditunjukkan pada Tabel 5.1. Dari tabel tersebut dapat disimpulkan bahwa panjang operator LLBP mempengaruhi proses *recognition* pada citra hasil ekstraksi fitur.

Jika semakin besar ataupun semakin kecil panjang operator LLBP, maka nilai akurasi akan menurun. Sedangkan nilai akurasi yang paling tinggi dihasilkan jika panjang operator LLBP adalah 13. Panjang operator LLBP yang optimal digunakan untuk *palm vein recognition* menggunakan metode LLBP yaitu 13. Citra hasil ekstraksi fitur dengan perbedaan panjang operator LLBP (N) dapat dilihat pada **Gambar 5.5**.



Gambar 5.5 Citra Hasil Ekstraksi fitur LLBP (a) Dengan N=11 (b) N=13 (c) N=15 (d) N=17 (e) N=19

5.5 Uji Coba Mendapatkan Akurasi pada *Finger Vein* menggunakan Metode Ekstraksi Fitur LLBP

Uji coba pengenalan pada jari tangan menggunakan deteksi pola pembuluh darah ini akan menghasilkan akurasi untuk setiap panjang operator LLBP. Tujuan dari uji coba ini yaitu untuk mengetahui performa metode LLBP pada proses ekstraksi fitur jika digunakan *recognition* pada setiap individu dengan perbedaan jumlah dan kualitas data testing. Kualitas citra yang bagus dilihat dari hasil *preprocessing* yang menunjukkan pola pembuluh darah lebih jelas.

Pengujian dilakukan dengan mendapatkan nilai akurasi pada setiap panjang operator LLBP. Pengujian dilakukan dengan menggunakan fungsi *similarity* pada MATLAB. Jika hasil *similarity* tinggi dan data testing sesuai dengan data training maka akan menambah nilai akurasi. Nilai panjang operator yang digunakan dalam uji coba adalah 11, 13, 15, 17, dan 19.

Tabel 5.2 Hasil Uji Coba Metode LLBP pada *Finger Vein*

FINGER VEIN LLBP	
Akurasi	
N	Data Testing
	35
11	68.57%
13	62.86%
15	62.86%
17	48.57%
19	40.00%

Hasil uji coba ditunjukkan pada Tabel 5.2. Dari tabel tersebut dapat disimpulkan bahwa panjang operator LLBP dan jumlah data testing mempengaruhi proses *recognition* pada citra hasil ekstraksi fitur. Jika semakin besar ataupun semakin kecil panjang operator LLBP, maka nilai akurasi akan menurun. Sedangkan nilai akurasi yang paling tinggi dihasilkan jika panjang operator LLBP adalah 11, yang berarti bahwa panjang operator tersebut merupakan panjang operator yang optimal untuk digunakan *finger vein recognition*. Karena panjang operator LLBP tersebut merupakan panjang operator yang standard [1].

5.6 Uji Coba Mendapatkan Akurasi pada *Palm Vein* menggunakan Metode Ekstraksi Fitur SIFT

Uji coba pengenalan pada telapak tangan menggunakan deteksi pola pembuluh darah ini akan menghasilkan akurasi untuk

setiap *threshold* pada proses *matching*. Tujuan dari uji coba ini yaitu untuk mengetahui performa metode SIFT pada proses ekstraksi fitur jika digunakan *recognition* pada setiap individu dengan perbedaan jumlah dan kualitas data testing. Kualitas citra yang bagus dilihat dari hasil *preprocessing* yang menunjukkan pola pembuluh darah lebih jelas.

Pengujian dilakukan dengan menggunakan fungsi *matching* pada MATLAB dari package *vl_feat*. Jika hasil *matching* tinggi dan data testing sesuai dengan data training maka akan menambah nilai akurasi. Nilai *threshold* yang digunakan dalam uji coba adalah 1.9, 2.1, 2.3, 2.5, 2.7, 2.9, 3.1 dan 3.3.

Tabel 5.3 Hasil Uji Coba Metode SIFT pada *Palm Vein*

PALM VEIN SIFT	
Akurasi	
Thresh	Data Testing
	35
1.9	51.43%
2.1	51.43%
2.3	60.00%
2.5	60.00%
2.7	60.00%
2.9	62.86%
3.1	62.86%
3.3	62.86%

Hasil uji coba ditunjukkan pada Tabel 5.3. Dari tabel tersebut dapat disimpulkan bahwa *threshold* pada proses *matching* mempengaruhi proses *recognition* pada citra hasil ekstraksi fitur. Selain *threshold*, data testing juga mempengaruhi proses *recognition*. Karena, dari data testing dan data training yang mempunyai intensitas cahaya, posisi, dan rotasi telapak tangan yang sama akan menghasilkan *similarity* yang tinggi.

5.7 Uji Coba Mendapatkan Akurasi pada *Finger Vein* menggunakan Metode Ekstraksi Fitur SIFT

Uji coba pengenalan pada jari tangan menggunakan deteksi pola pembuluh darah ini akan menghasilkan akurasi untuk setiap *threshold* pada proses *matching*. Tujuan dari uji coba ini yaitu untuk mengetahui performa metode SIFT pada proses ekstraksi fitur jika digunakan *recognition* pada setiap individu dengan perbedaan jumlah dan kualitas data testing. Kualitas citra yang bagus dilihat dari hasil *preprocessing* yang menunjukkan pola pembuluh darah lebih jelas.

Pengujian dilakukan dengan menggunakan fungsi *matching* pada MATLAB dari package *vl_feat*. Jika hasil *matching* tinggi dan data testing sesuai dengan data training maka akan menambah nilai akurasi. Nilai *threshold* yang digunakan dalam uji coba adalah 1.9, 2.1, 2.3, 2.5, 2.7, 2.9, 3.1 dan 3.3.

Tabel 5.4 Hasil Uji Coba Metode SIFT pada *Finger Vein*

FINGER VEIN SIFT	
Akurasi	
Thresh	Data Testing
	35
1.9	82.86%
2.1	82.86%
2.3	85.71%
2.5	80.00%
2.7	80.00%
2.9	74.29%
3.1	80.00%
3.3	80.00%

Hasil uji coba ditunjukkan pada Tabel 5.4. Dari tabel tersebut dapat disimpulkan bahwa *threshold* pada proses *matching*

mempengaruhi proses *recognition* pada citra hasil ekstraksi fitur. Selain *threshold*, kualitas data juga mempengaruhi proses uji coba pada metode ini. Dari hasil uji coba metode SIFT pada data *finger vein* menunjukkan bahwa *threshold matching* yang paling optimal untuk digunakan dalam *finger vein recognition* menggunakan metode SIFT yaitu 2.3. Selain itu kualitas data yang digunakan dalam uji coba yaitu rotasi jari dan intensitas cahaya juga mempengaruhi proses *matching*.

5.8 Evaluasi

Subbab ini akan dijelaskan hasil dari serangkaian uji coba yang dilakukan dan kendala yang dihadapi selama proses pengerjaan. Evaluasi yang dilakukan adalah pada tahap *preprocessing*, tahap ekstraksi fitur yaitu menggunakan metode LLBP dan SIFT, dan pada tahap *matching* untuk data *finger vein*.

5.8.1 Evaluasi pada Tahap *Preprocessing*

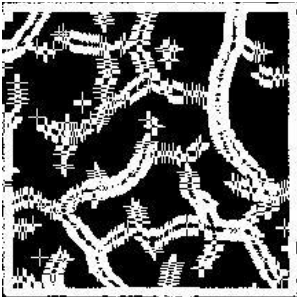
Tahap *preprocessing* pada tugas akhir ini diterapkan pada dua dataset yaitu pada citra *palm* dan *finger vein*. Pada *preprocessing palm vein* mengacu pada referensi *preprocessing* untuk metode ekstraksi fitur LLBP, sedangkan untuk *preprocessing* pada *finger vein* mengacu pada referensi *preprocessing* untuk metode ekstraksi fitur SIFT. Dilihat dari hasil uji coba yang telah dilakukan, hasil akurasi untuk *palm vein* menggunakan metode ekstraksi fitur LLBP lebih bagus dibanding dengan hasil akurasi untuk *finger vein* menggunakan metode ekstraksi fitur LLBP. Selain itu, intensitas cahaya citra juga mempengaruhi proses *preprocessing*, sehingga hasil *preprocessing* setiap citra juga bergantung pada kualitas citra.

5.8.2 Evaluasi pada Tahap Ekstraksi Fitur

Proses ekstraksi fitur menggunakan dua metode yaitu LLBP dan SIFT. Parameter yang digunakan dalam ekstraksi fitur dua metode tersebut berbeda. Berikut masing-masing penjelasan evaluasi hasil uji coba:

5.8.2.1 Ekstraksi Fitur menggunakan Metode LLBP

Proses ekstraksi fitur menggunakan metode LLBP, terdapat satu parameter yang diuji coba untuk mengetahui performa dari metode LLBP jika diimplementasikan pada data *palm* maupun *finger vein* yaitu panjang operator LLBP. Fitur LLBP berupa matriks 2 dimensi yang dapat direpresentasikan dengan citra. Untuk parameter panjang operator LLBP akan mempengaruhi hasil ekstraksi fitur. Semakin besar nilai panjang operator LLBP maka pola pembuluh darah akan semakin terlihat lebar pada citra. Berikut contoh representasi fitur LLBP dan matriks citra tersebut.



Gambar 5.6 Representasi Fitur LLBP dengan Panjang Operator LLBP 11

216x216 double											
	1	2	3	4	5	6	7	8	9	10	11
1	43.9090	0	0	130	156.0128	0	248.0081	0	0	96	
2	0	58.0345	100.0200	58.0345	18	0	86	220.0091	32.0624	140.0143	96.0
3	0	142.6184	56.6039	17.0880	96.0208	64.4050	64.2806	118.0678	87.2009	144.8862	97.5
4	140.0143	102	76.2365	160.4244	98.4886	76.0263	154.2077	96.7471	65.1153	27.2029	114.0
5	0	24.0832	129.3213	93.3809	121.4907	86.9253	175.4537	142.3938	52.8394	108.8118	25.2
6	120	180.0111	66.4831	138.6795	75.2861	82.4621	249.6237	227.6928	186.0108	48	135.2
7	118	96.0208	187.9255	189.4518	138.7516	199.4492	147.5127	123.1747	162.6899	179.7443	179.7
8	0	74.0270	184.6943	97.3242	141.2091	80	291.3280	285.8041	139.3126	207.4608	96.3
9	86	60.0333	161.5549	182.4829	75.8947	148.1891	159.0220	91.2140	60.7289	110.0545	137.1
10	44	44.0454	6	34.9285	63.8122	138.6362	153.6750	196.4586	153.1535	107.6290	39.3
11	94	90.0222	88.4081	78.7147	64.8999	120.7477	114.9087	64	104.1729	34.1760	113.7
12	126	62.0322	106.6208	58.3095	83.5225	98.1835	118.3385	79.6241	82	73.4302	136.9
13	130	126.0159	10	50.9902	42.1900	66.2420	133.1465	149.6262	112.6410	88	116.4

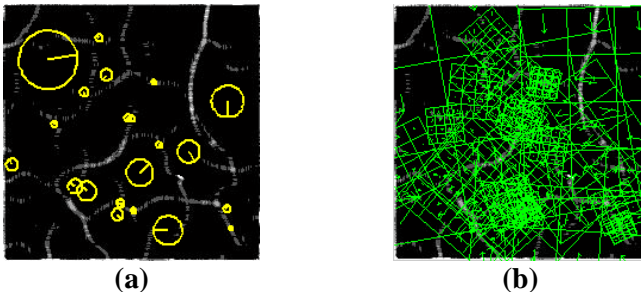
Gambar 5.7 Matriks Hasil Ekstraksi Fitur LLBP dengan Panjang Operator 11

Pada **Gambar 5.7** merupakan nilai hasil ekstraksi fitur LLBP. Setiap piksel mempunyai nilai yang merupakan fitur LLBP dari citra asli, fitur LLBP bisa direpresentasikan kedalam citra seperti pada **Gambar 5.6**. Nilai terendah dari fitur LLBP yaitu 0, sedangkan untuk nilai tertinggi dari fitur LLBP berdasarkan parameter panjang operator LLBP.

Metode ekstraksi fitur LLBP jika diimplementasikan pada dataset *finger vein* akan menghasilkan akurasi yang rendah. Akurasi yang rendah tersebut disebabkan oleh beberapa faktor yaitu kualitas dataset yang kurang bagus (intensitas cahaya rendah) dan posisi finger pada data *scan*.

5.8.2.2 Ekstraksi Fitur menggunakan Metode SIFT

Proses ekstraksi fitur menggunakan metode SIFT, terdapat satu parameter yang diuji coba untuk mengetahui performa dari metode SIFT jika diimplementasikan pada data *palm* maupun *finger vein*. Parameter tersebut yaitu *threshold* yang digunakan pada proses *matching*. Citra yang mempunyai intensitas cahaya rendah akan mempengaruhi hasil deteksi *keypoint* dan deskriptor yang berbeda-beda, sehingga akan mempengaruhi hasil akurasi. Setiap citra akan menghasilkan *keypoint* dan deskriptor yang berbeda beda. *Keypoint* dan deskriptor dapat direpresentasikan seperti **Gambar 5.8**.



Gambar 5.8 Fitur SIFT (a)*Keypoint* (b)*Deskriptor*

Metode ekstraksi fitur SIFT jika diimplementasikan pada dataset *finger vein* akan menghasilkan akurasi yang lebih tinggi dibanding jika diimplementasikan pada dataset *palm vein*. Akurasi

yang rendah tersebut disebabkan oleh beberapa faktor yaitu kualitas dataset yang kurang bagus (intensitas cahaya rendah).

5.8.3 Evaluasi pada Tahap *Matching* untuk data *Finger Vein*

Proses *matching* untuk data *finger vein* menggunakan library dari Matlab. Fungsi ini mempunyai tiga parameter yaitu nilai deskriptor untuk data testing, nilai deskriptor untuk data training, dan *threshold*. Nilai *threshold* akan mempengaruhi hasil *matching* antar citra. Selain *threshold*, hasil ekstraksi fitur berupa *keypoint* dan deskriptor juga akan mempengaruhi hasil *matching*. Karena jumlah *keypoint* pada setiap citra bergantung pada kualitas citra tersebut.

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah. Selain itu juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

6.1 Kesimpulan

Kesimpulan yang diperoleh dari uji coba dan evaluasi adalah sebagai berikut:

1. Metode ekstraksi fitur LLBP lebih cocok digunakan pada dataset *palm vein*, sedangkan metode ekstraksi fitur SIFT lebih cocok digunakan pada dataset *finger vein*.
2. Panjang operator LLBP yang optimal dilakukan pada proses ekstraksi fitur yaitu diantara 11 sampai 13. Metode ekstraksi fitur LLBP jika diimplementasikan pada *palm vein* menghasilkan akurasi rata-rata 85.72%. Sedangkan jika diimplementasikan pada dataset *finger vein* menghasilkan akurasi rata-rata 56.57%.
3. Data testing yang digunakan untuk uji coba pada *palm* maupun *finger vein* akan mempengaruhi proses *recognition*, yang mengakibatkan akurasi uji coba menjadi rendah jika kualitas citra rendah. Sebaliknya, jika kualitas citra tinggi akurasi yang dihasilkan juga tinggi. Untuk uji coba *palm vein* menggunakan metode SIFT menghasilkan akurasi rata-rata yaitu 58.93%. Sedangkan hasil akurasi rata-rata pada data *finger vein* adalah 80.72%. Sehingga dapat disimpulkan bahwa, *finger vein recognition* lebih cocok menggunakan metode ekstraksi fitur SIFT.

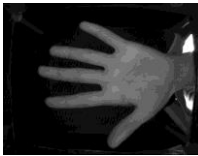
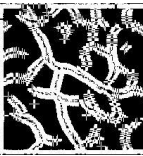
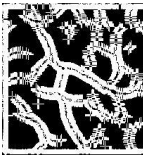
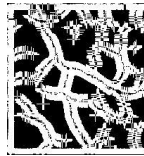
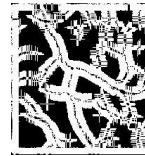
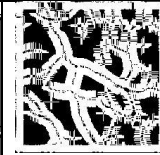
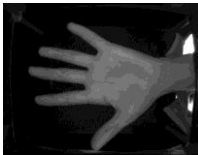
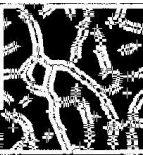
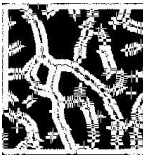

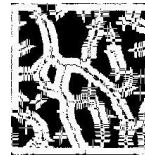
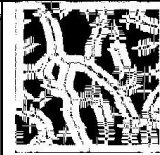


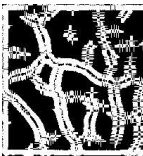
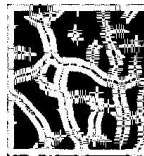
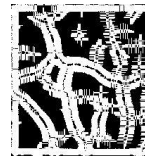
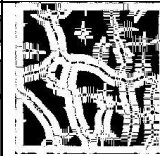
6.2 Saran

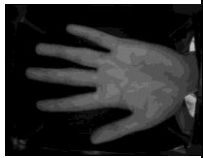
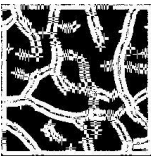
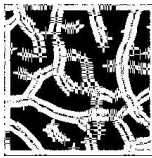
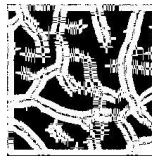
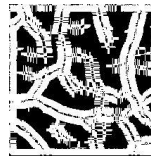
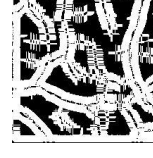
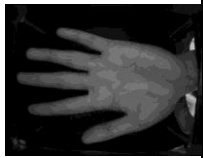
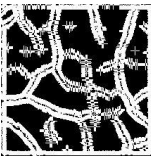
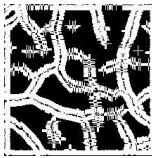
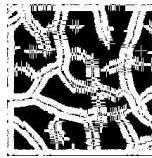
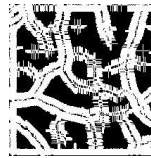
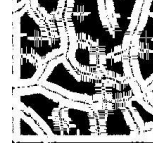
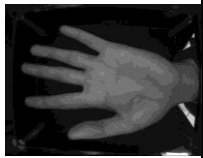
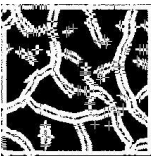
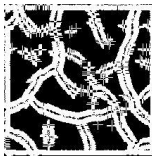
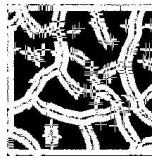
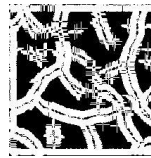
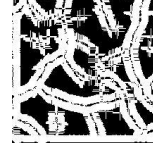
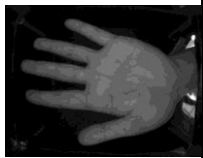
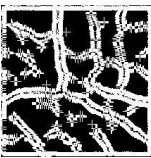
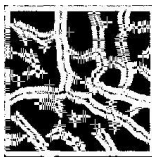
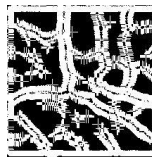
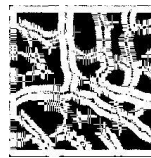
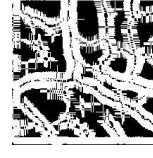
Saran yang dapat diberikan dalam pengujian metode ekstraksi fitur LLBP dan SIFT pada *palm* dan *finger vein* adalah sebagai berikut:

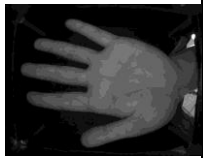
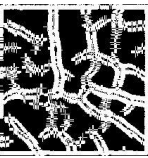
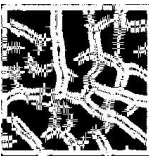
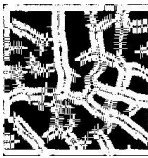
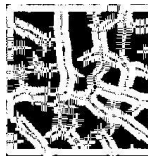

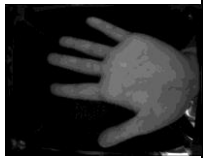
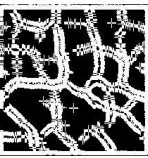
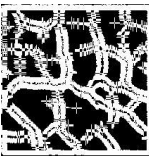
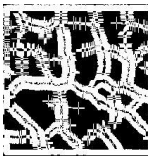
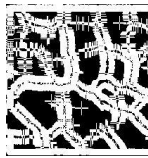
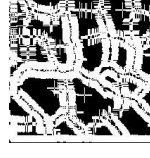
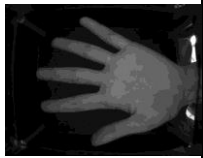
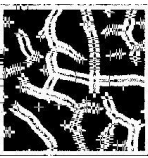
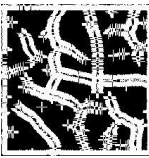
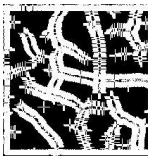
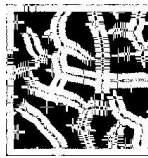
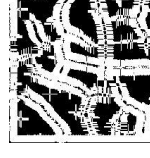
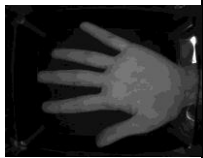
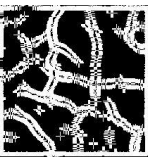
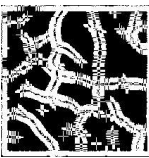
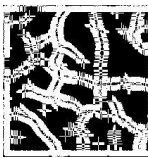
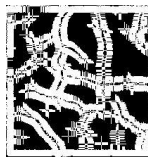
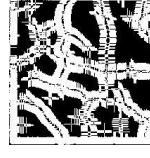
1. Perbaiki *preprocessing* untuk memperjelas pola pembuluh darah pada *palm* maupun *finger vein* agar menghasilkan data fitur yang lebih representatif.

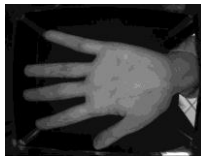
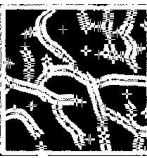
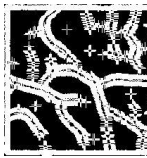
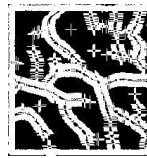
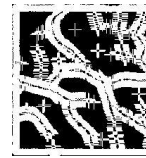
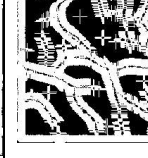

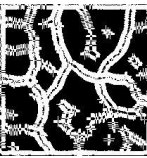
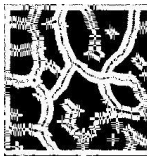
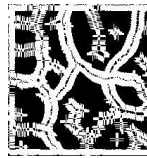
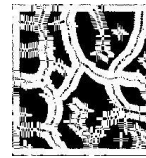

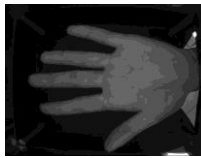
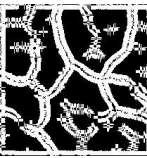
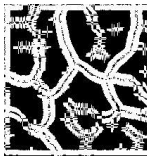
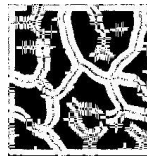
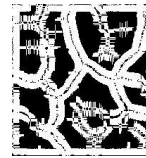
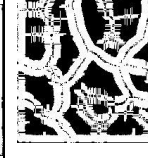


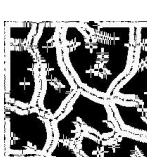
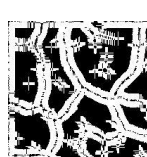

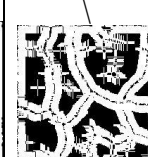
LAMPIRAN

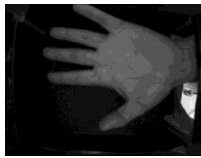
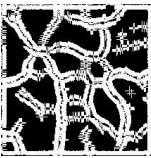
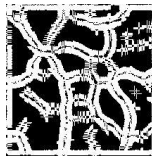
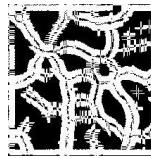
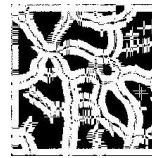

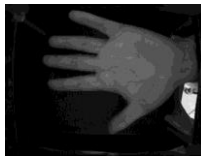
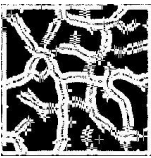
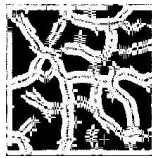
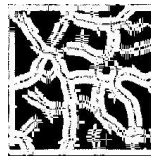
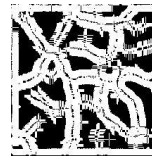


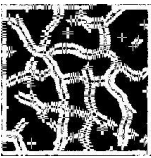
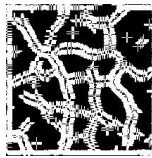
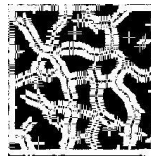
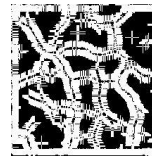
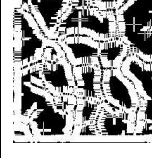
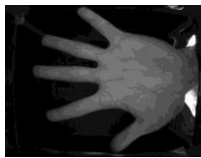
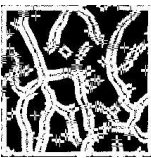
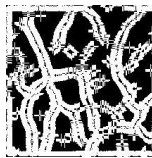
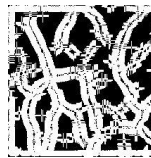
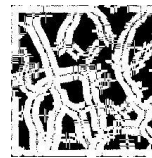
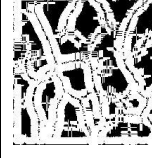
A. Hasil Ekstraksi Fitur data *training* menggunakan Metode LLBP pada *Palm Vein*

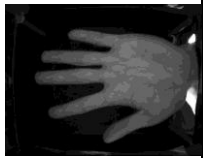
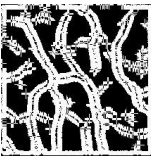
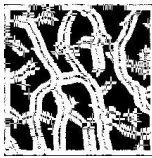
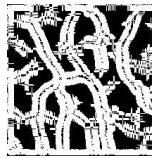
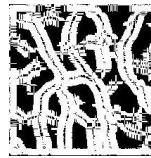
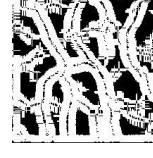

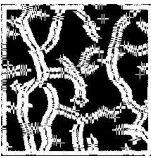
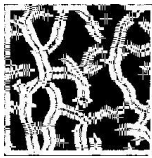
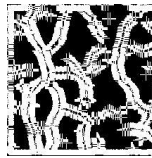
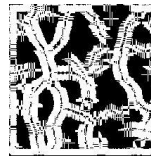
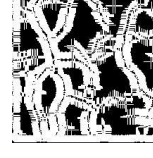
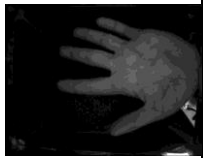
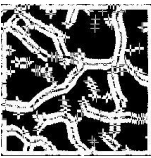
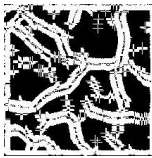
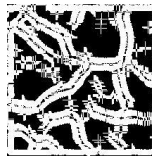
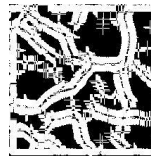
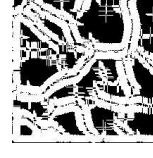
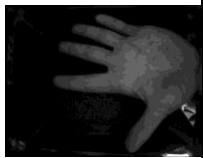
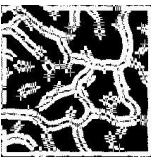
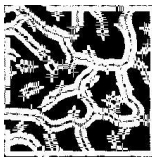
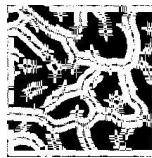
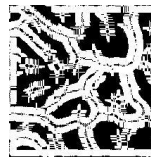
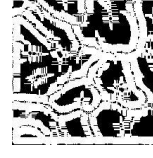
Nama Citra	Citra Asli	N				
		11	13	15	17	19
005_1_940_02						
005_1_940_03						
005_1_940_04						

007_1_940_ 02						
007_1_940_ 03						
007_1_940_ 04						
009_1_940_ 02						

009_1_940_ 03						
009_1_940_ 04						
010_1_940_ 02						
010_1_940_ 03						

010_1_940_04						
014_1_940_02						
014_1_940_03						
014_1_940_04						

017_1_940_ 02						
017_1_940_ 03						
017_1_940_ 04						
019_1_940_ 02						

019_1_940_ 03						
019_1_940_ 04						
021_1_940_ 02						
021_1_940_ 03						

[Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] J. Y. Sari, C. Faticah, and N. Suciati, "LOCAL LINE BINARY PATTERN FOR FEATURE EXTRACTION ON PALM VEIN RECOGNITION," *J. Ilmu Komput. Dan Inf.*, vol. 8, no. 2, pp. 111–118, Aug. 2015.
- [2] H. Khusnuliawati, "Multi-feature Fusion menggunakan Fitur Scale Invariant Feature Transform dan Local Extensive Binary Pattern untuk Pengenalan Pembuluh Darah pada Jari," Tesis, ITS Surabaya, 2016.
- [3] "Palm vein recognition based on multi-sampling and feature-level fusion," *ResearchGate*. [Online]. Available: https://www.researchgate.net/publication/269728973_Palm_vein_recognition_based_on_multi-sampling_and_feature-level_fusion. [Accessed: 04-Jun-2017].
- [4] M. Watanabe, "Palm vein authentication technology and its applications," Sep. 2005.
- [5] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," Jan. 2004.
- [6] "APLIKASI PENGENALAN PALM VEIN BERDASARKAN FITUR GABO." [Online]. Available: http://rbtc.if.its.ac.id/v4/index.php?p=show_detail&id=11383. [Accessed: 04-Jun-2017].
- [7] "Contrast-limited adaptive histogram equalization (CLAHE) - MATLAB adapthisteq." [Online]. Available: <https://www.mathworks.com/help/images/ref/adapthisteq.html>. [Accessed: 04-Jun-2017].
- [8] A. Rachmad, "PENGOLAHAN CITRA DIGITAL MENGGUNAKAN TEKNIK FILTERING ADAPTIVE NOISE REMOVAL PADA GAMBAR BERNOISE," Nov. 2008.
- [9] "2-D adaptive noise-removal filtering - MATLAB wiener2." [Online]. Available:

- <https://www.mathworks.com/help/images/ref/wiener2.html>.
[Accessed: 04-Jun-2017].
- [10] “2-D median filtering - MATLAB medfilt2.” [Online].
Available:
<https://www.mathworks.com/help/images/ref/medfilt2.html>.
[Accessed: 04-Jun-2017].
- [11] “Morphologically close image - MATLAB imclose.”
[Online]. Available:
<https://www.mathworks.com/help/images/ref/imclose.html>.
[Accessed: 04-Jun-2017].
- [12] “Morphologically open image - MATLAB imopen.”
[Online]. Available:
<https://www.mathworks.com/help/images/ref/imopen.html>.
[Accessed: 04-Jun-2017].
- [13] “Structural similarity,” *Wikipedia*. 10-May-2017.
- [14] “VLFeat - Tutorials > SIFT detector and descriptor.”
[Online]. Available:
<http://www.vlfeat.org/overview/sift.html>. [Accessed: 04-Jun-2017].

BIODATA PENULIS



Lusiana Nurul Aini, lahir di Rembang, pada tanggal 16 Januari 1996. Penulis menempuh pendidikan mulai dari TK Hasyimiyah (2000-2001), SDN 1 Sedan (2001-2007), SMPN 1 Sedan (2007-2010), SMAN 1 Rembang (2010-2013) hingga dilanjutkan di Institut Teknologi Sepuluh Nopember Surabaya (2013-2017) di jurusan Teknik Informatika, Fakultas Teknologi Informasi angkatan tahun 2013.

Selama belajar di kampus Teknik Informatika, penulis berkesempatan menjadi administrator Laboratorium Pemrograman 2. Selain mengikuti kegiatan akademik, penulis mengikuti kegiatan organisasi sekaligus mengembangkan minatnya pada bidang *public relation* dengan berpartisipasi sebagai staff Departemen Hubungan Luar HMTC (2014-2015), staff Organization Social Responsibility BEM FTIf (2014-2015), dan Sekretaris Departemen Hubungan Luar HMTC (2015-2016). Selain itu penulis juga aktif dalam kepanitiaan *Big Event* HMTC yaitu SCHEMATICS (2014-2016).

Penulis memiliki bidang minat Komputasi Cerdas Visi (KCV) dengan fokus studi pada bidang *image processing*. Komunikasi dengan penulis dapat melalui email: **lusiananurulaini@gmail.com**.